



# Contextually aware intelligent control agents for heterogeneous swarms

Adam J. Hepworth<sup>1</sup> · Aya S. M. Hussein<sup>1,3</sup> · Darryn J. Reid<sup>1,2</sup> · Hussein A. Abbass<sup>1</sup>

Received: 20 November 2022 / Accepted: 25 January 2024  
© Crown 2024

## Abstract

An emerging challenge in swarm shepherding research is to design effective and efficient artificial intelligence algorithms that maintain simplicity in their decision models, whilst increasing the swarm's abilities to operate in diverse contexts. We propose a methodology to design a context-aware swarm control intelligent agent (shepherd). We first use swarm metrics to recognise the type of swarm that the shepherd interacts with, then select a suitable parameterisation from its behavioural library for that particular swarm type. The design principle of our methodology is to increase the situation awareness (i.e. contents) of the control agent without sacrificing the low computational cost necessary for efficient swarm control. We demonstrate successful shepherding in both homogeneous and heterogeneous swarms.

**Keywords** Context-aware swarm · Information markers · Swarm analytics · Swarm control · Shepherding

## 1 Introduction

Contemporary approaches to swarm guidance and control often assume that swarm agents are homogeneous in their response to external influence vectors. This manifests in the design of control algorithms, such as herding, often operating directly on the raw positional

---

✉ Adam J. Hepworth  
a.j.hepworth@unsw.edu.au

Aya S. M. Hussein  
aya.hussein@canberra.edu.au

Darryn J. Reid  
darryn.reid@dst.defence.gov.au

Hussein A. Abbass  
h.abbass@unsw.edu.au

<sup>1</sup> School of Systems and Computing, University of New South Wales, Northcott Drive, Canberra, ACT 2600, Australia

<sup>2</sup> Defence Science and Technology Group, Adelaide, SA 5111, Australia

<sup>3</sup> School of Information Technology and Systems, University of Canberra, Bruce, ACT 2617, Australia

data of swarm agents to compute influence vectors, where simple interaction rules may not require complex assessments to determine optimal behaviours. Herding-based models, such as shepherding, have been implemented for over 25 years, with classic control methods typically operating on simple transformations of raw data (Hasan et al., 2022). Swarm shepherding is an example of a swarm control herding-based method where one or more external actuators (sheepdogs) operate on low-level information by calculating primitive statistical features from raw data. These models often use static behaviour selection policies for the control agent to guide a swarm to a goal location (Debie et al., 2021). As a biologically inspired approach to swarm control, shepherding has applications across different domains, such as the guidance and control of crowds (Lee and Kim et al., 2012), herding biological animals (Paranjape et al., 2018), guiding teams of uncrewed system (UxS) (Hepworth, 2021), and controlling a group of robotic platforms (Lee & Kim, 2017; Cowling and Gmeinwieser, 2010).

Classic swarm intelligence systems rely on reactive, mostly memoryless, models. Reactivity implies that the agent does not use sophisticated cognitive functions such as predicting future system states, projecting the consequence of actions, or planning to make a decision. Instead, reactive models sense the environment and use simple functions to directly project sensed information onto actions. They are memoryless when they only rely on immediate sensed information to create their actions; technically, we say they fulfil the first-degree Markovian assumption where immediate transitions depend only on the current states. These characteristics make reactive models computationally efficient and act significantly faster than cognitive models, which attempt to add more sophistication in the processing layers between sensors and actuators. The simplicity of reactive models, though, comes with drawbacks as they are agnostic to contextual and situational circumstances and fail when the complexity in the environment increases (El-Fiqi et al., 2020). Common examples of when they fail include the presence of noise in sensed data and increased diversity (heterogeneity) in the swarm contexts they need to operate within.

There are common nested sets of assumptions amongst most swarm control models. Examples of these assumptions include cases where sensor data are restricted to binary, restricting decision models to linear cases, limiting the continuity of the dynamics to event-based simulations, considering all agents in the swarm to be homogeneous, and assuming that all parameters are known in advance (Long et al., 2020). Furthermore, the most prevalent approaches to shepherding employ rule-based policies, with relatively few examples of swarm control systems that allow policies to switch behaviours (Hussein et al., 2022), see, for example, Go et al. (2021), Zhi and Lien (2021), and Debie et al. (2021). Whilst the simplicity of reactive models offers many benefits in their application, more complex individual- and team-level behaviours could be achieved by improving agent's perception without necessarily complicating an agent's decision-making model (Bredeche & Fontbonne, 2022).

Few studies consider heterogeneous agents, particularly swarms where behaviours of the collective and individuals may change over time (Hepworth et al., 2020), or employ context-aware approaches (Abbass et al., 2018). Still, the predominant research direction follows work such as Strömbom et al. (2014), where members of the swarm are assumed to be homogeneous and there is a single sheepdog (actuator-agent) guiding the swarm. Swarms containing heterogeneous agents are known to increase swarm control complexity and unpredictability (Özdemir et al., 2017).

When agents in the swarm exhibit a wide variety of behaviours, the control agent needs to be able to recognise these behaviours to deploy the right decision-making model for the particular situation it is facing. In other words, the control agent needs to be able to

recognise and classify situations and contexts. Such an ability will still allow the agent to use a reactive model. However, the model could get parameterised differently or could be structurally different in different contexts. The recognition of these contexts is an open research problem.

Recognising distinct contexts and group dynamics is crucial as one agent's actions can have a significant impact on the entire group (Jolles et al., 2020), limiting the effectiveness of the control agent. A control agent needs to effectively categorise the profile of members and the overall dynamics of the swarm, select an appropriate response/tactic, and act on this tactic. The primary research question we ask in this manuscript is: *how can context be used to adapt behaviour?* Specifically, we would like to design a methodology that allows the control agent to recognise swarm characteristics, then select the most effective herding tactics to guide the swarm. Our objective is to integrate context-awareness into the decision-making process for a rule-based swarm control agent. Our hypothesis is that by recognising a context and parameterising the control strategy accordingly, the control agent will operate successfully in more situations.

Rule-based shepherding is often criticised for its inability to address dynamic and unknown swarm contexts; we address this research gap by designing a context-aware system. The control agent is extended with two abilities for context recognition and tactic selection. We first evaluate the performance of various herding strategies across a range of homogeneous and heterogeneous swarms in a shepherding context. We seek to answer the question: *how do different herding strategies impact mission performance across different shepherding contexts?* In the second phase, we consider the context-aware adaptation of the control agent's behaviour by monitoring the swarm in real-time to select a herding strategy based on the inferred swarm characteristics. Put simply, we want to know *how can context be used to influence behaviour selection?*

Our previous work Hepworth et al. (2023) designed a set of information markers for swarm analytics. The contribution was at the level of context analysis and identification without using these markers to improve the decision making of the control agent. These shortfalls form the main contributions of the current manuscript. In particular, the contributions of this paper are as follows:

1. We extend our previous work on information markers with new metrics for evaluating swarm performance. The combined set of metrics is then used to categorise the different types of agents in a swarm. We then present a classifier that takes the information markers and outputs a behavioural category of agents.
2. We present a methodology to systematically assess the relationship between the behavioural categories of agents and performance under diverse shepherding behaviours in both homogeneous and heterogeneous swarm scenarios. The methodology leads to the design of a mapping between the situation and state of the swarm and the control policy that needs to be adopted to that situation.
3. The previous two methodologies formed the basis for designing a library of situations and behaviours to support the control agent to guide the swarm. The library approach allows the control agent to be reactive, whilst the reactive model used could change based on the real-time analysis of the situation.

In Sect. 2, we review approaches to swarm control and influence, surveying essential model-type formulations and critically assessing the underlying assumptions of these models. Then, in Sect. 3, we present our context-aware system and formulate the problem

space. Next, Sects. 4 and 5 cover the systematic analysis of the problem space to derive the sensor data into actionable-information (S2AI), followed by the actionable-information to actuation (AI2A) design that transforms that analysis into context-aware actions. Finally, we conclude the paper in Sect. 6 with a discussion on future research opportunities for context-aware swarm control.

## 2 Swarm control

Swarm control encompasses a wide range of approaches to swarming, typically categorised as centralised or decentralised, with examples of each in both simulations and on physical systems (Long et al., 2020). The most prominent algorithms to swarm control are based on the notions of influence zones utilising reactive decision models, typically characterised with spatial features (Vicsek et al., 1995). Biologically inspired approaches feature prominently within the literature, for instance, the natural flocking of birds (Reynolds, 1987), the predator–prey interactions of sheep and sheepdogs (Strömbom et al., 2014), and hunting methods of a wolf pack (Hu et al., 2022).

The current approaches to swarm control often focus on problems where swarms of agents possess homogeneous sensors, reactive decision processes, and limited action sets (Hepworth et al., 2020). Centre of mass, also known as centre of gravity, -based approaches are a standard method to control a swarm, with few assumptions around agent homogeneity or influence distribution within the swarm. The current approaches are often data-driven, operating on spatially derived features to calculate the next move of an agent, for instance, Mohamed et al. (2021) and El-Fiqi et al. (2020). Biologically inspired models use derivations from empirical research as the basis for agent parameterisations.

One approach to centralised control is shepherding, “inspired by sheepdogs and sheep, where the shepherding problem can be defined as the guidance of a swarm of agents from an initial location to a target location” (p. 523) (Long et al., 2020). Proposals to solve the shepherding challenge include bio-inspired algorithms, heuristic-based rule algorithms, and machine learning solutions, including neural networks and reinforcement learning approaches (Hasan et al., 2022). The predominant approach to extensions of rule- or heuristic-based methods employs arc, line, or circle formations, usually relying on the centre of mass and exact position of agents to conduct guidance. Few studies research the impact of limited sensing ranges and the use of only local information by the control agent, for instance, Mohamed et al. (2021) and El-Fiqi et al. (2020).

### 2.1 Shepherding

Shepherding behaviours use an external control agent to guide the swarm. Strömbom et al. (2014) introduce a shepherding model of interactions between a flock of sheep and a sheepdog based on empirical field trial data. The heuristic model employs a self-propelled particle approach to reproduce the attraction and repulsion interactions of the two agent types. The model describes  $N$  swarm agents (sheep,  $\pi$ ) placed within an  $L \times L$  area (paddock) with  $M$  control agents (sheepdogs,  $\beta$ ). Swarm agent behaviour is generated as a force vector that combines attraction to their local centre of mass, repulsion from other  $\pi$  and repulsion from  $\beta$  agents. The task of  $\beta$  agents is to move  $\pi$  agents to a particular location, a goal area. Two behaviours are presented for the control, being *collect* and *drive*, derived from side-to-side movements introduced by Lien et al. (2004). The control agent collects

members of the swarm within a radius ( $f(N)$ ) and then drives the swarm towards the goal location. The mission is considered complete when the swarm centre of mass is within a distance  $\delta$  from the goal, signifying that the flock is within the target goal area.

These usually operate on the raw positional data of swarm agents, or simple transformations of these, relying on diverse control models for herding. For example, Cowling and Gmeinwieser (2010) employ a hierarchical and stack-based finite state machine. Fujioka (2017) discusses a hybrid control method based on positional data that combines formation and collection control methods. The method utilises a V-formation control to guide a V-shaped notch towards a goal location, where collecting via centre (CvC) sees the shepherding agent collect with one of the three position-cases, being left, right, or centre. Tsunoda et al. (2018) demonstrate that aiming for the furthest sheep to the goal produces a more superior performance to the one proposed by Vaughan et al. and Strombom et al.

Zhang and Pan (2022) propose a control agent employing a distributed collecting algorithm that does not require the centre of mass to be calculated and that drives using a density-based method instead of using the convex hull of the flock. The approach has robustness over classic methods, which may be able to address adversarial agents, such as non-cooperative members or threats external to the swarm. Auletta et al. (2022) propose a set of local control rules for a small group of herding agents to collect and herd a swarm. The primary difference in the model proposed considers the situation where some swarm agents do not possess the ability to cooperate with other members of the swarm (unable to flock), increasing the complexity of the problem space for the swarm control agent. The solution developed is implemented as a distributed approach where each herder agent selects a strategy based on local feedback, driving their decision selection for what targets to follow.

Varadharajan et al. (2022) consider a variation of the shepherding problem in which a particular pattern is maintained during movement to the goal, noted as crucial for applications such as nanomedicine or smart materials or where spatial configurations may have functional implications. Also considered is that the shepherds can modulate interaction forces between the sheep. Himo et al. (2022) consider the response where some swarm agents are unresponsive to the herding agent. These heterogeneous traits of some swarm members are parameterised through modification of an agent's repulsion forces to the control agent. An unresponsive agent is encoded with a lower response weight than the standard swarm agent, and a non-responsive agent repulsion force coefficient is set to zero.

Özdemir et al. (2017) address physical space and energy constraints through a method that offers a reduced compute requirement, minimising the information collected and processed by individual agents. Hu et al. (2022) address challenges of flexibility and efficiency when herding large numbers of sheep, proposing a novel coordination protocol that requires fewer herding agents to complete the task. Hu et al. note that this foundation presents an opportunity to extend this work in real-world environments and further optimise protocol parameters.

Existing rule-based algorithms lack adaptability to respond in changing environments (obstacles) or with changing swarms (non-homogeneous); however, they are shared, in part, due to their simplicity and potential application in real robotic systems (Zhang et al., 2022). Recent shepherding models consider using learning-based control algorithms, addressing some shortfalls of rule-based algorithms. For example, Zhi and Lien (2021) propose a method of shepherding to herd agents amongst obstacles using a deep reinforcement learning approach. Hussein et al. (2022) consider curriculum-based reinforcement learning and propose an algorithm that demonstrates superior performance to that of the classic rules-based agent. Mohamed et al. (2021) introduce a graph-based approach that

promotes cohesion between swarm members, improving shepherding performance and mission success outcomes.

## 2.2 Critical assessment of swarm control methods

Common amongst methods identified throughout the survey of Long et al. (2020) is the use of similar control model formulations. These include that the control agent decision model is often rule-based, initiating primitive force-vector behaviours based on raw data or simple transformation of this. Long et al. highlight that the values of the weights for different force vectors depend on context. He further goes on to discuss that modulation of these weights may be required in different environments structures or in settings where uncertainty exists in either the sensing or decision-action output of an agent. For contexts where the swarm agents need increased intelligence, the agents need to be equipped with the skills to recognise the activities they observe in the contexts they operate within. Research on activity recognition is missing with the swarm literature. A discussion on heterogeneous swarms is absent. Long et al. (2020) indicate that future shepherding control systems could be complemented by expanding on the current methods with additional capabilities, such as goal planning and path planning, as well as considering new behavioural sets to enhance dynamics complexity.

It is difficult to select and/or know when to execute those behaviours that will lead to effective shepherding, especially that the swarm is operating under increased uncertainty. Predominantly, proposed control methods assume that the control agent has access to perfect information for the position of the swarm agents (Lee & Kim 2017). In this paper, we require adaptive behaviours to cater for uncertainty in the swarm behaviour and the state; a summary of models and swarm control implementations is contained in Table 1.

In simple settings, limited cognition elements are required, and the swarm control agent can often act on the raw positional information. However, in settings where the homogeneity assumption is relaxed, enhanced cognition of the swarm control agent may be required to determine agent characteristics and understand how these manifest as a source of control imbalance in the swarm. This point of differentiation in approaches to typical explorations of swarm control models allows us to consider a more comprehensive range of scenarios that could include adversarial agents, additional collaborating control agents, and environmental complexities.

Zhang et al. (2022) highlight that typical herding patterns require two behaviours: collecting and driving. A central assumption is that swarm agents are collected and, once aggregated, driven towards a goal location. Focussing on the task execution sequence may introduce fragility to reactive shepherding models, limiting the possible strategies for a control agent to implement. In settings where the swarm constituent agents possess heterogeneous properties, it may be desirable to collect and drive sub-groups of agents, or one at a time, to the goal location. In such a scenario, the swarm may only be collected at the goal location as the final agent arrives.

The above review suggests an open research gap for recognising swarm and swarm agent characteristics. Without such a capability, a swarm control agent cannot understand the cumulative impacts of influences on and in a swarm and subsequently act on this information to determine the most appropriate control strategy. Furthermore, this approach requires integrating contexts to develop increased situational awareness.

**Table 1** Summary of studies discussed, highlighting the consideration of model setting, decision model type, swarm heterogeneity, and control agent context-awareness. Not all studies focus on swarm control; however, all employ a model of swarming within their solution. Rule-based solutions are those typically with precise mathematical formulations

Study	Setting	Decision model	Framework	Heterogeneous	Context-aware
Reynolds (1987)	Flocking	Rule-based	Decentralised		
Vicsek et al. (1995)	Flocking	Rule-based	Decentralised		
Lien et al. (2004)	Shepherding	Rule-based	Single agent		
Lien et al. (2005)	Shepherding	Rule-based	Multi-agent		
Cowling and Gmeinwieser (2010)	Shepherding	Finite state machine	Single agent		
Lee and Kim et al. (2012)	Shepherding	Rule-based with path planning	Single agent		
Strömboom et al. (2014)	Shepherding	Rule-based	Single agent		
Maschian and Royan (2015)	Shepherding	Rule-based	Decentralised		
Kengyel et al. (2015)	Honeybee	Evolutionary	Decentralised	✓	
Zhang et al. (2016)	Flocking	Adaptive controller	Decentralised	✓	
Lee and Kim (2017)	Shepherding	Rule-based	Multi-agent		
Özdemir et al. (2017)	Shepherding	Rule-based	Single agent		
Jang et al. (2018)	Fish school	Markov chain	Decentralised	✓	
Hoshi et al. (2018)	Shepherding	Rule-based	Single agent		
Tsunoda et al. (2018)	Shepherding	Rule-based	Single agent	✓	
Goel et al. (2019)	Shepherding	Rule-based	Multi-agent		
Chen et al. (2019)	Flocking	Rule-based	Decentralised	✓	
Singh et al. (2019)	Shepherding	Rule-based	Single agent		✓
Traboulsi and Barbeau (2019)	Non-bio	Soft-max regression	Decentralised		✓
Nguyen et al. (2019)	Shepherding	Deep reinforcement learner	Single agent		
Hepworth et al. (2020)	Shepherding	Rule-based	Single agent	✓	
Mohamed et al. (2021)	Shepherding	Connectivity-aware rule-based	Single agent		
El-Fiqi et al. (2020)	Shepherding	Rule-based	Multi-agent		
Mohanty et al. (2020)	Non-bio	Deep learning	Decentralised	✓	

Table 1 (continued)

Study	Setting	Decision model	Framework	Heterogeneous	Context-aware
Go et al. (2021)	Shepherding	Imitation learning	Single agent		✓
Zhi and Lien (2021)	Shepherding	Reinforcement learner (RL)	Single agent		
Debie et al. (2021)	Shepherding	Neuro-evolutionary NN-RL	Single agent		
Zhang and Pan (2022)	Shepherding	Rule-based	Multi-agent		
Auletta et al. (2022)	Shepherding	Rule-based	Multi-agent		
Varadharajan et al. (2022)	Shepherding	Rule-based	Multi-agent		
Himo et al. (2022)	Shepherding	Rule-based	Single agent	✓	
Hussein et al. (2022)	Shepherding	Reinforcement learner	Single agent		
Zhang et al. (2022)	Shepherding	Rule-based	Multi-agent		
Abpekar et al. (2022)	Flocking	Rule-based	Distributed		
Hu et al. (2022)	Wolf Pack	Rule-based	Multi-agent		



### 3 Context-aware intelligent system

Pajares Ferrando and Onaindia (2013) define three requirements for a context-aware system: information extraction, interpretation of extracted information, and ability to adapt system's functionality to current use. Our motivation for a context-aware system is to provide a control agent with the ability to select and modulate autonomous behaviours for swarm control. Often rule-based swarm control agents receive information from the environment that leads to a new autonomous behaviour being selected. For example, a swarm agent could alternate between collect and drive actions when one sheep is zigzagging around the threshold for considering this sheep as an astray one. These cases can result in a decision *deadlock* in which the control agent continuously changes its implemented behaviour, leading to limited or no progress towards the mission goal. Our idea is to integrate time-based behaviour modulations to maintain performance stability. We hypothesise that the combination of time-based behaviour modulation and contextual awareness will increase performance and decrease observed instability across the swarm in control settings.

In our previous work Hepworth et al. (2023), an AI observer aimed to reveal one or more hidden states of the swarm. In our current work, the observer's role extends to supporting the shepherding agent by providing contextual information to parameterise the shepherding agent's behaviours for an observed situation. The observer perceives the environment to infer the context, characterising the type of agents and swarm. Upon classifying the context, the context-aware system parameterises the behaviours available to the control agent to best guide the swarm to the goal location. The context-aware system constrains the autonomous behaviours available to the control agent, modulating its impact on the swarm. The context-awareness system observes the environment for a fixed window to detect changes in the situation.

Figure 1 conceptualises our extension of the work presented in (Fig. 2) by bringing together the primary categories of concepts used by the observer to reason.

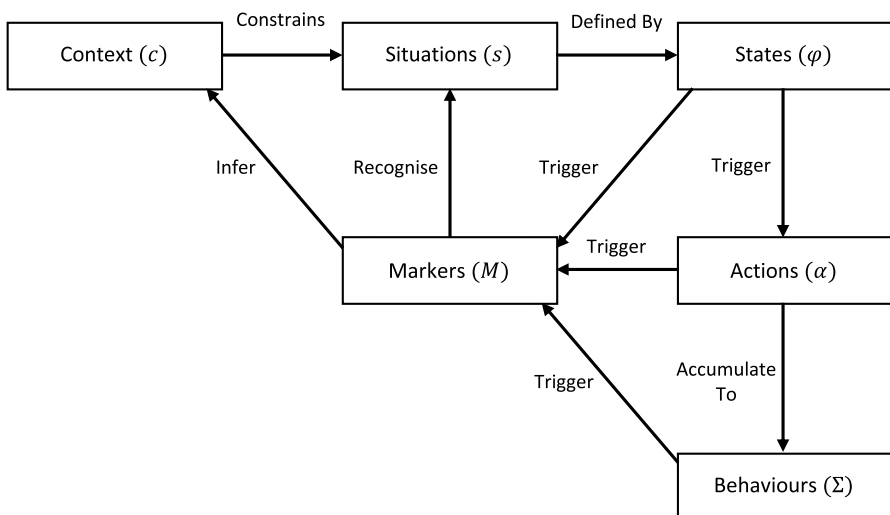
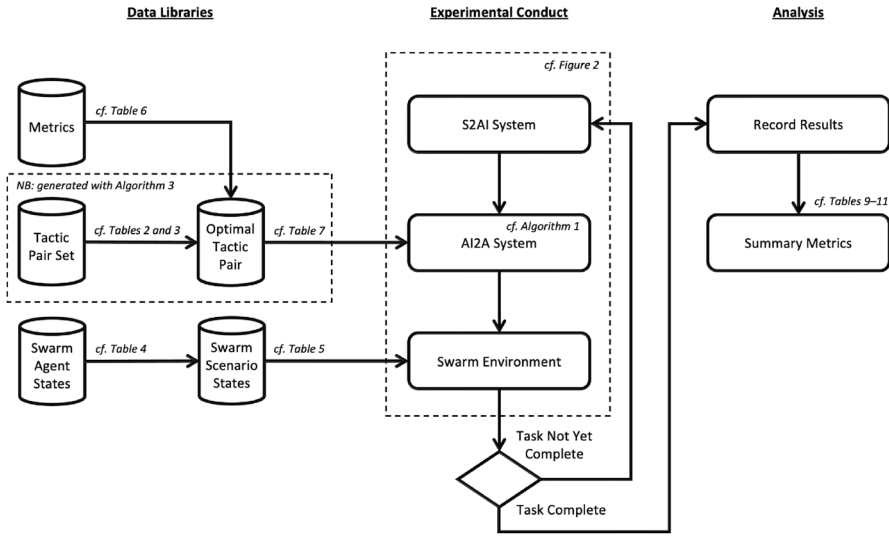
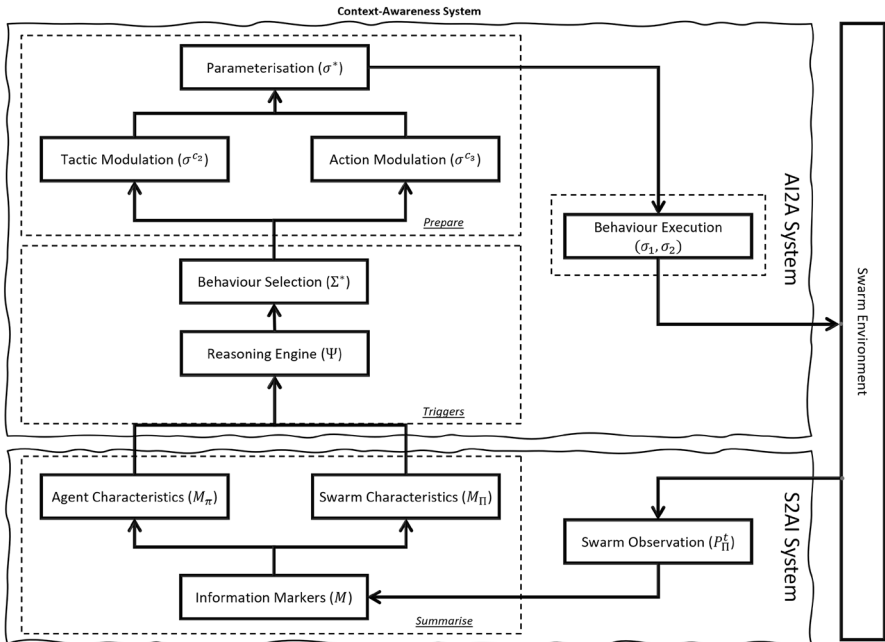


Fig. 1 Conceptual linkages of definitions, expanded on that presented in Hepworth et al. (2023)



**Fig. 2** Overview of the methodology schema. The S2AI system observes the swarm and develops actionable-information, with the AI2A system acting on this information within the swarm environment and moving agents to the goal location. The AI2A system core is implemented as per Algorithm 1. The left hand side of this figure, data libraries, represents the offline system setup, whilst the middle (experimental conduct) and right hand side (analysis) represent the online use of context information



**Fig. 3** Conceptual architecture depicting the flow of information for the context-awareness system presented

The reasoning architecture for the context-aware system is presented in Fig. 3, the methodology is depicted in Fig. 2. The context-aware system goes through two phases: sensor data to actionable-information phase (S2AI) and actionable-information to actuation phase (AI2A). Our previous work in Hepworth et al. (2023) offered the indicators used in S2AI, whilst the contribution of this paper is the AI2A phase.

The proposed architecture augments a classic shepherding agent with cognitive capabilities to operate above a reactive level such as the one described in Strömbom et al. (2014). The two phases function as follows:

- *Sensor data to actionable-information phase (S2AI)*
  - *Summarise* The *summarise* function is based on the information markers method presented in Hepworth et al. (2023). Information markers ( $\mathcal{M}$ ) use position information of all  $\pi_i$  agents to derive state information on  $\Pi$ . As described in Fig. 1, we use the information markers as features for a classifier to recognise contexts. Marker states uncover particular aspects of agent's characteristics ( $\mathcal{M}_{\pi_i}$ ). Individual states get aggregated to categorise the swarm ( $\mathcal{M}_{\Pi}$ ), for instance, the distribution of agents and type of homogeneity or heterogeneity observed.
- *Actionable-Information Phase to Actuation (AI2A)*
  - *Trigger* The *trigger* function fuses individual and collective characteristics about the swarm to enable the reasoning engine ( $\Psi$ ) to infer situations and contexts. The reasoning engine longitudinally takes the output of the classifier as an input to estimate the most likely situation that the swarm is facing. We then use the estimated situation to select behaviours for the control agent ( $\Sigma^*$ ).
  - *Prepare* The *prepare* function determines how often an autonomous behaviour should be executed ( $\sigma^{c_2}$ ), as well as how frequently the selected behaviour should be re-parameterised ( $\sigma^{c_3}$ ). The modulation of these behaviour elements enables the control agent to consider multiple periods of swarm response, designed to address the often high levels of sensorial noise. The final aspect of the prepare function is to parameterise the control agent with the behaviours available for execution ( $\sigma^*$ ) and the constraints for their employment ( $\sigma^{c_2}, \sigma^{c_3}$ ).

Algorithm 1 describes the functions depicted in Fig. 2. The system has two user-defined inputs: the observation window length ( $\omega$ ) and the proportion of overlap between two successive context windows ( $\tau \in [0, 1]$ ), as given on Line 1. In our previous work, we optimised these parameters, where  $\omega = 60$  and  $\tau = 0.75$  Hepworth et al. (2023). The system continues to operate until the mission success criterion is achieved; that is, the distance between the goal location and the centre of mass of the swarm is below a threshold,  $\|P_{\Pi} - P_G\| \leq R_G$  (Line 2).

During the observation period window ( $\Delta\omega$ ), the context-awareness system observes the coordinate position of each  $\pi$ , given as  $P_{\pi_i}^{\Delta\omega}$  (Lines 3–4). These recorded observations are used to first calculate the information markers ( $\mathcal{M}$ , Line 5) for each  $\pi$ , classifying each agent as a particular type ( $A$ , Line 6), as given in Table 2. Finally, each probabilistic classification is summarised as a state vector ( $A^*$ ) that reports that the likelihood each agent type has is observed; the state vector  $A^*$  is the empirical swarm distribution (Line 8).

For each known scenario in Table 3 ( $s \in S$ , Line 9), we then calculate the  $L_2$ -norm distance between  $s$  and  $A^*$ , given as  $\|s - A^*\|_2$  (Line 10). We employ a modified inverse distance weighting procedure inspired by the work of Shepard (1968) to estimate the

**Table 2** Summary of agent state vector weight parameterisations

Agent state	$W_{\pi\Lambda}$	$W_{\pi\pi}$	$W_{\pi\beta}$	$s_{\pi}/s_{\beta}$
A1	0.50	2.00	0.50	1.00
A2	1.50	2.00	0.50	0.50
A3	0.50	3.00	1.00	0.67
A4	0.50	2.00	1.90	0.67
A5	1.05	3.00	1.00	0.67
A6	1.05	1.50	1.00	0.50
A7	1.05	2.00	1.00	0.67

Each agent is parameterised with an agent state as defined, with distinct homogeneous and heterogeneous groupings used through different scenarios

**Table 3** Summary of swarm scenarios used in this study, as originally presented in Hepworth et al. (2023)

Scenario state	A1	A2	A3	A4	A5	A6	A7
S1	0.20						0.80
S2		0.20	0.20			0.20	0.40
S3				0.80			0.20
S4	0.20				0.20		0.60
S5							1.00
S6	1.00						
S7		1.00					
S8			1.00				
S9				1.00			
S10					1.00		
S11						1.00	

Each swarm state highlights the distribution of agent types present, across both homogeneous and heterogeneous scenarios

likelihood that our observed  $A^*$  is scenario  $s$ , generating a probability likelihood state vector for all  $S$  (Line 11). We calculate the swarm markers state ( $\|\mathcal{M}^{\Pi}\|_2$ , Line 13) using the same technique for  $\mathcal{M}^{\Pi}$  and probabilistically classify the marker state as a particular swarm type (Line 14). We combine the outputs of this and the previous method into an ensemble to classify the most likely observed swarm scenario ( $\hat{\Pi}$ ). We select  $\hat{\Pi}^*$  by selecting the most likely scenario overall context windows observed from  $t = 0$  to the current observation period, considering probabilistic variability. In practical terms, we select the scenario that returns the maximum likelihood as  $\arg \max(\text{mean}(\hat{\Pi}) - \text{var}(\hat{\Pi}))$  (Line 15).

We conduct pair-wise statistical tests to determine the significance of  $\hat{\Pi}^*$ , comparing to all other  $\hat{\Pi}$  (Line 16). For the case where  $\hat{\Pi}^*$  is significant (i.e. has statistically higher likelihood than all  $\hat{\Pi}$ ) and the tactic pair (TP,  $\mathfrak{Z}$ ) currently employed is significant (Line 17), then we parameterise  $\beta$  with  $\mathfrak{Z}$  (Line 18). In the case where  $\mathfrak{Z}$  is not assessed as significant, then we select a new  $\mathfrak{Z}$  from  $\Sigma$  library that maximises the likelihood of mission success, for the given scenario (Line 20). In the case where  $\hat{\Pi}^*$  is not assessed as significant (i.e. does not have a higher likelihood than  $\hat{\Pi}$ ), we select the highest likelihood scenario type as a two-class problem between heterogeneous or homogeneous (Line 23) and parameterise  $\mathfrak{Z}$  in either of these forms (i.e. not for a specific scenario but a class of scenarios, Line 24). Our final tasks are to determine  $\sigma^{c_2}$  (Line 26) and  $\sigma^{c_3}$  (Line 27) from  $\hat{\Pi}^*$  and  $\mathfrak{Z}$ , then parameterise  $\beta$  for the next context period to influence the swarm (Line 28).

### 3.1 Multi-agent reasoning for swarm control

Our system architecture consists of three agent types: the context-aware observer agent, cognitive agent, and shepherding agent. Each agent is independent and specialises in a single task; we have discussed the context-awareness system and now describe cognitive and shepherding agent functions.

- *Observer agent* The observer agent is responsible for sensing the environment, deriving the information markers, and classifying the situation facing the agent. It then provides parameterised behaviours to the cognitive agent.
- *Cognitive agent* In the architecture, the cognitive agent determines which behaviour from the context-aware agent's parameters should be conducted, selecting between  $\sigma_1$  and  $\sigma_2$ , consistent with prior shepherding models. The function of the cognitive agent is to calculate  $f(N)$  to select a behaviour,  $\sigma^*$ , to be conducted, determined to be either  $\sigma_1$  or  $\sigma_2$ . The cognitive agent makes this decision with the parameterised behaviours provided by the context agent and the positions of all swarm agents in the environment.
- *Shepherding agent* The shepherding agent is the actuator that delivers force vectors to influence the swarm to move to the goal location. The shepherding agent decides on a behaviour  $\sigma^*$  and how to conduct it, calculating the position ( $P_{\beta_s}^t$ ) at which the agent will deliver the influence vector associated with its selected behaviour. The function of the shepherding agent is to calculate the behaviour position point and execute a force vector to influence the swarm based on the behaviour selected and the configuration of the swarm.

Algorithm 1. Context-aware decision shepherding agent ( $\omega, \tau$ )

---

```

1: Set context window length  $\omega$  and context window overlap  $\tau$ 
2: while  $\|P_{\Pi} - P_G\| > R_G$  do
3:   Observe  $P_{\pi_i}^{\Delta\omega} \quad \forall \pi_i \in \Pi$ 
4:   for  $\pi_i \in \Pi$  do
5:     Calculate  $\mathcal{M}^{\pi_i}$ 
6:     Classify  $\hat{\pi}_i : \mathcal{M}^{\pi_i} \rightarrow A$ 
7:   end for
8:   Summarise  $A^* = \text{mean}(A)$   $\triangleright \forall A$ 
9:   for  $s \in S$  do
10:    Calculate  $\|s - A^*\|_2$ 
11:    Calculate inverse weighting distance to obtain  $Pr(A^*) \rightarrow s$ 
12:   end for
13:   Calculate  $\|\mathcal{M}^{\Pi}\|_2$ 
14:   Classify  $\hat{\Pi} : \mathcal{M}^{\Pi} \rightarrow S$ 
15:    $\hat{\Pi}^* = \arg \max(\text{mean}(\hat{\Pi}) - \text{var}(\hat{\Pi}))$ 
16:   if  $\hat{\Pi}^*$  significant then
17:     if  $\mathfrak{T}^t$  significant then
18:       Continue  $\mathfrak{T}^t$ 
19:     else
20:       Select  $\arg \max(\text{TP} \rightarrow s^*)$ 
21:     end if
22:   else
23:     Classify  $\hat{\Pi}^* \rightarrow S = \arg \max(s_{He}, s_{Ho})$ 
24:     Set  $\mathfrak{T}$  to  $S^*$ 
25:   end if
26:   Calculate  $\sigma^{c2} \sim \hat{\Pi}^*, \mathfrak{T}$ 
27:   Calculate  $\sigma^{c3} \sim \hat{\Pi}^*, \mathfrak{T}$ 
28:   Parameterise  $\beta = \langle \sigma^{c2}, \sigma^{c3}, \mathfrak{T} \rangle$ 
29: end while

```

---

#### 4 S2AI: Sensor to actionable-information

This section aims to examine the effectiveness of different *collect* and *drive* behaviours in Strömbom et al. (2014) on the success of shepherding. We introduce two points of departure from the original model: the modulation of the control behaviours available to the shepherding agent and the inclusion of heterogeneous swarm agents. Our intent in systematically varying these aspects of the model is to investigate the performance of distinct combinations of autonomous control behaviours (collect and drive actions), across different homogeneous and heterogeneous swarm scenarios.

#### 4.1 Shepherding agent behaviours

A single instantiation of each behaviour is usually presented in the majority of shepherding-based models, for instance, Lien et al. (2004) and Strömbom et al. (2014), often activated through a linear rule-based switching mechanism that operates on low-level positional information of each of the swarm agents. This study introduces an expanded set of five collect and five drive behaviours. The driving position for the drive behaviour ( $\sigma_1$ ) as given in Strömbom et al. (2014) is calculated by the function

$$P_{\beta\sigma_1}^t = \Lambda_{\beta}^t + f(N) \cdot \frac{\Lambda_{\beta}^t - P_G^t}{\|\Lambda_{\beta}^t - P_G^t\|}, \quad (1)$$

where  $f(N) = R_{\pi\pi} N^{2/3}$ .  $P_{\beta\sigma_1}^t$  is the position for agent  $\beta$  to execute behaviour  $\sigma_1$  at time  $t$ ;  $\Lambda_{\beta}^t$  is the local centre of mass for the swarm to be controlled by agent  $\beta$  at  $t$ ; and  $\Lambda_{\beta}^t - P_G^t / \|\Lambda_{\beta}^t - P_G^t\|$  is the direction in which the driving point needs to be oriented. The driving behaviour is triggered if the following inequality, signifying that the swarm is well-grouped, is satisfied.

$$\forall \pi_i \in \Omega_{\beta\pi}^t, \|\Lambda_{\beta^i}^t - P_{\pi}^t\| \leq f(N), \quad (2)$$

where  $\Omega_{\beta\pi}^t$  is the set of  $\pi$  agents that the  $\beta$  agent operates on. We summarise the algorithm by Strömbom et al. (2014) in Algorithm 2.

##### Algorithm 2. Reactive shepherding agent

---

```

1: while  $\|P_{\Pi} - P_G\| > R_G$  do  $\triangleright R_G$  is the goal radius
2:   Observe  $P_{\pi_i}^t \forall i$ 
3:   Set  $\Omega_{\beta\pi}^t = \min\{\Omega^t, \Omega_{\beta\pi}\}$  of  $\pi_i$  agents to  $\beta$ 
4:   Set  $\Lambda_{\beta}^t$  as the centre of mass
5:   Calculate  $f(N) = R_{\pi\pi} N^{2/3}$ 
6:   if  $\forall \pi_i \in \Omega_{\beta\pi}^t, \|\Lambda_{\beta^i}^t - P_{\pi}^t\| \leq f(N)$  then
7:     Conduct driving behaviour,  $\sigma_1$ 
8:   else  $\exists \pi_i^* \in \Omega_{\beta\pi}^t, \arg \max_i (\|\Lambda_{\beta}^t - P_{\pi_i}^t\| > f(N))$ 
9:     Conduct collecting behaviour,  $\sigma_2$ 
10:  end if
11: end while

```

---

Note that in Algorithm 2, the inequality for  $f(N)$  is calculated at every time step. We modify the term  $f(N)$  by including a threshold  $\mathcal{L}$  that modulates  $N$ . This serves as a constraint on the number of agents  $\beta$  operates on to determine whether  $\Pi$  is clustered for the purpose of driving. The resulting inequality is now given as

$$\forall \pi_i \in \Omega_{\beta\pi}^t, \|\Lambda_{\beta^i}^t - P_{\pi}^t\| \leq f(\mathcal{L}N), \quad (3)$$

where we set  $\mathcal{L} = \{1.00, 0.75, 0.50, 0.25, 1/N\}$  to generate the additional four behaviours. We abbreviate each behaviour as drive-100 (D100,  $\mathcal{L} = 1.00$ ), drive-75 (D75,  $\mathcal{L} = 0.75$ ), drive-50 (D50,  $\mathcal{L} = 0.50$ ), and drive-25 (D25,  $\mathcal{L} = 0.25$ ). The final drive behaviour is a special case that we call drive-one (D1N,  $\mathcal{L} = 1/N$ ), being the selection of a single agent for driving. The collecting point behaviour ( $\sigma_2$ ) can be given by the function



$$P_{\beta\sigma_2}^t = P_{\pi_i^*}^t + R_{\pi\pi} \cdot \frac{P_{\pi_i^*}^t - \Lambda_{\beta}^t}{\|P_{\pi_i^*}^t - \Lambda_{\beta}^t\|}. \tag{4}$$

The attention of the swarm agent is focussed on the agent furthest from the swarm, activated by the inequality.

$$\exists \pi_i^* \in \Omega_{\beta\pi}^t, \arg \max_i (\|\Lambda_{\beta}^t - P_{\pi_i}^t\| > f(N)). \tag{5}$$

We introduce additional collect reference points to generate four different behaviours, modifying the selection of a single  $\pi_i$  to collect. The collecting behaviours available to the shepherding agent include:

- select the  $\pi_i$  closest to the  $\beta$  shepherding agent (C2D, closest-to-dog),
- select the  $\pi_i$  closest to the herd of sheep,  $\Pi$ , (C2H, closest-to-herd centre of mass),
- select the  $\pi_i$  furthest to  $\beta$  shepherding agent (F2D, furthest-to-dog),
- select the  $\pi_i$  furthest to goal position,  $P_G$ , (F2G, furthest-to-goal), or
- select the  $\pi_i$  furthest to the herd of sheep,  $\Pi$  (F2H, furthest-to-herd centre of mass).

Algorithm 3 summarises changes to Algorithm 2. The key changes between these algorithms include:

1. initialising  $\sigma_1$  to an element of the set {D100, D75, D50, D25, D1} and
2. initialising  $\sigma_2$  to an element of the set {C2D, C2H, H2D, F2G, F2H}.

Algorithm 3. Multiple-behaviour reactive shepherding agent ( $\sigma_1^i, \sigma_2^j \sigma_{i1}, \sigma_{j2}$ )

---

```

1: while  $\|P_{\Pi} - P_G\| > R_G$  do
2:   Observe  $P_{\pi_i}^t \forall i$ 
3:   Set  $\Omega_{\beta\pi}^t = \min\{\Omega^t, \Omega_{\beta\pi}\}$  of  $\pi_i$  agents to  $\beta$ 
4:   Set  $\Lambda_{\beta}^t$  as the centre of mass
5:   Calculate  $f(N) = R_{\pi\pi} N^{2/3}$ 
6:   if  $\forall \pi_i \in \Omega_{\beta\pi}^t, \|\Lambda_{\beta}^t - P_{\pi_i}^t\| \leq f(N)$  then
7:     Conduct driving behaviour,  $\sigma_1$ 
8:   else based on the collect reference point,
9:     Conduct collecting behaviour,  $\sigma_2$ 
10:  end if
11: end while

```

---

We parameterise the primitive behaviours of collect and drive for the control agent, implemented in a behaviour library and summarised in Table 4.

Our shepherding model instantiates a one collect and one drive behaviour employed by  $\beta$  to control  $\Pi$ . We denote a pair of behaviour as a *tactic pair* (TP); that is,  $TP = \{\sigma_1, \sigma_2\}$ ; in the remainder of this paper, we use a unique TP label instead of behaviour combinations using such as  $\{\sigma_1^{D100}, \sigma_2^{F2H}\}$ . Our objective with all TPs is to evaluate the combinations of collect and drive behaviours, consisting of 25 TPs ( $TP_{1,\dots,25}$ ). Table 5 introduces the parameterisations of collect and drive behaviours in this study.

## 4.2 Swarm heterogeneity

Similar to Hepworth et al. (2023), we adopt three weights for  $\pi$ -agents and a speed differential. Specifically, we vary

- $W_{\pi\Lambda}$ : strength of attraction for a  $\pi$  to their local centre of mass,  $\Lambda$ .
- $W_{\pi\pi}$ : strength of repulsion for a  $\pi$  to another  $\pi$ .
- $W_{\pi\beta}$ : strength of repulsion for a  $\pi$  to the shepherding agent  $\beta$ .
- $\frac{s_\pi}{s_\beta}$ : speed differential between a  $\pi$  agent and a  $\beta$  agent.

The seven  $\pi_i$  agent types and parameterised weights in this study are given in Table 2, as originally presented in Hepworth et al. (2023). The parameterisation of agent A7 is as initially presented by Strömbom et al. (2014).

The  $\pi$ -type agents are allocated to different swarms as the constituent members of 11 scenarios, consisting of four heterogeneous and seven homogeneous swarms, presented in Table 3 of Hepworth et al. (2023) and summarised here in Table 3. Strömbom et al. (2014) employ swarm exclusively S5 in their study, a homogeneous scenario with agent type A7. We include an additional seven homogeneous swarms scenarios (S5, S6, S7, S8, S9, S10, S11), one for each agent type, as shown in Table 2. The scenarios represent the different contexts that the control agent must recognise and act upon. We also parameterise four heterogeneous swarm scenarios (S1, S2, S3, S4), as first presented in Hepworth et al. (2023).

Note that the swarm agent behaviour model remains unchanged, ensuring that the low-computation principle of swarm robotics is maintained. The swarm agents do not possess knowledge of the particular context or other swarm agent interactions rules, responding per individual parameterisations.

## 4.3 Experimental design and analysis

Our experimental design aims to evaluate each TP's performance in both homogeneous and heterogeneous swarm settings. We assess the performance of each TP for a particular scenario 30 times, resulting in a total of 8,250 trials (30 trials for 25 TPs in 11 scenarios); 20  $\pi$  agents and 1  $\beta$  agent were used similar to Debie et al. (2021). In addition, we collect a range of summary statistics on TP performance, summarised by the metrics presented in Table 6.

Rizk et al. (2019) discuss the development of standardised evaluation metrics, highlighting that "the evaluation criteria include specific domain performance metrics and domain invariant criteria", offering examples such as spatiotemporal, complexity, load, fairness, communication, robustness, scalability, and resource-based metrics. Selecting appropriate evaluation metrics depends on the desired characteristics of the model at hand, which can include characteristics such as autonomy, complexity, adaptability, concurrency, distribution, and communication (Shehory & Sturm 2001).

Various methods have been proposed to evaluate the performance of swarm control methods. These methods often compare the performance of one control approach to another or to understand the sensitivity of particular parameter settings; for instance, see variations of Strömbom et al. (2014) such as El-Fiqi et al. (2020), Zhang et al. (2022), and Singh et al. (2019). However, measures to evaluate the inclusion of cognitive capabilities as outlined

**Table 4** Summary of control agent behaviours parameterised in this study

Behaviour	Type	Nomenclature	Description
D1N	Drive	$\sigma_1^{D1N}$	Drive exactly one agent of the herd
D25	Drive	$\sigma_1^{D25}$	Drive at least $\frac{1}{4}$ of the herd
D50	Drive	$\sigma_1^{D50}$	Drive at least $\frac{1}{2}$ of the herd
D75	Drive	$\sigma_1^{D75}$	Drive at least $\frac{3}{4}$ of the herd
D100	Drive	$\sigma_1^{D100}$	Drive all of the herd
C2D	Collect	$\sigma_2^{C2D}$	Closest swarm agent to shepherding agent
C2H	Collect	$\sigma_2^{C2H}$	Closest swarm agent to the herd centre of mass
F2D	Collect	$\sigma_2^{F2D}$	Furthest swarm agent to shepherding agent
F2G	Collect	$\sigma_2^{F2G}$	Furthest swarm agent to the goal
F2H	Collect	$\sigma_2^{F2H}$	Furthest swarm agent to the herd centre of mass

**Table 5** The set of 25 tactic pairs used in this study systematically explores five variations of each  $\sigma_1$  and  $\sigma_2$ . TP<sub>5</sub> (denoted with \*) represents the behaviour introduced by (Strömbom et al. 2014).

Tactical pairs		Collect behaviour				
		C2D	C2H	F2D	F2G	F2H
Drive behaviour	D100	$\{\sigma_1^{D100}, \sigma_2^{C2D}\} \rightarrow TP_1$	$\{\sigma_1^{D100}, \sigma_2^{C2H}\} \rightarrow TP_2$	$\{\sigma_1^{D100}, \sigma_2^{F2D}\} \rightarrow TP_3$	$\{\sigma_1^{D100}, \sigma_2^{F2G}\} \rightarrow TP_4$	$\{\sigma_1^{D100}, \sigma_2^{F2H}\} \rightarrow TP_5$
	D75	$\{\sigma_1^{D75}, \sigma_2^{C2D}\} \rightarrow TP_1$	$\{\sigma_1^{D75}, \sigma_2^{C2H}\} \rightarrow TP_2$	$\{\sigma_1^{D75}, \sigma_2^{F2D}\} \rightarrow TP_3$	$\{\sigma_1^{D75}, \sigma_2^{F2G}\} \rightarrow TP_4$	$\{\sigma_1^{D75}, \sigma_2^{F2H}\} \rightarrow TP_5$
	D50	$\{\sigma_1^{D50}, \sigma_2^{C2D}\} \rightarrow TP_1$	$\{\sigma_1^{D50}, \sigma_2^{C2H}\} \rightarrow TP_2$	$\{\sigma_1^{D50}, \sigma_2^{F2D}\} \rightarrow TP_3$	$\{\sigma_1^{D50}, \sigma_2^{F2G}\} \rightarrow TP_4$	$\{\sigma_1^{D50}, \sigma_2^{F2H}\} \rightarrow TP_5$
	D25	$\{\sigma_1^{D25}, \sigma_2^{C2D}\} \rightarrow TP_1$	$\{\sigma_1^{D25}, \sigma_2^{C2H}\} \rightarrow TP_2$	$\{\sigma_1^{D25}, \sigma_2^{F2D}\} \rightarrow TP_3$	$\{\sigma_1^{D25}, \sigma_2^{F2G}\} \rightarrow TP_4$	$\{\sigma_1^{D25}, \sigma_2^{F2H}\} \rightarrow TP_5$
	D1	$\{\sigma_1^{D1}, \sigma_2^{C2D}\} \rightarrow TP_1$	$\{\sigma_1^{D1}, \sigma_2^{C2H}\} \rightarrow TP_2$	$\{\sigma_1^{D1}, \sigma_2^{F2D}\} \rightarrow TP_3$	$\{\sigma_1^{D1}, \sigma_2^{F2G}\} \rightarrow TP_4$	$\{\sigma_1^{D1}, \sigma_2^{F2H}\} \rightarrow TP_5$

**Table 6** Metrics used in this study and interpretations for tactic pair assessments

Metric	Equation	Interpretation
Mission success	$MS = \begin{cases} 1 & \text{if } P_{\Pi}^T - P_G^T < R_G \\ 0 & \text{otherwise} \end{cases}$	Effectiveness metric for the overall mission success. Over multiple trials, the objective is to maximise the number of successes
Mission completion rate	$MCR = \frac{\ P_{\Pi}^T - P_G^T\ }{\ P_{\Pi}^T - P_{\Pi}^{t=0}\ }$	Efficiency metric for the shepherding agent's control strategy. A lower mission completion rate indicates increased performance for this metric
Mission speed	$MSp = \frac{\ P_{\Pi}^T - P_{\Pi}^{t=0}\ }{\Delta t}$	Efficiency metric measuring the pace of mission execution. Higher mission speed indicates increased performance
Mission decision stability	$MDS = \frac{MS}{1 + \sum_{j=1}^T \chi_1(\sigma(t), \sigma(t+1))}$	Sensitivity metric informing on the mission impact to changes in the number of control decisions. Higher mission decision stability indicates increased performance for this metric
Decision swarm stability	$DSS = \frac{1/ \Pi  \sum_{j=1}^{ \Pi } \chi_2(\pi_j, \Pi^*)}{1 + \sum_{j=1}^T \chi_1(\sigma(t), \sigma(t+1))}$	Sensitivity metric informing if a control strategy could be considered susceptible to perturbations in change. Lower decision swarm stability indicates increased performance for this metric
Mission swarm stability	$MSS = \frac{MS}{1 + 1/ \Pi  \sum_{j=1}^{ \Pi } \chi_2(\pi_j, \Pi^*)}$	Sensitivity metric informing on the mission impact to changes in swarm stability. Higher mission swarm stability indicates increased performance for this metric

A single metric is reported for each simulation trial as a scenario summary, calculated at each simulation step

by Long et al. (2020) are yet to be widely established, highlighting the requirement for an evaluation approach to account for the contribution of new capabilities within the agent.

To address this gap, we adopt six new metrics. Table 6 summarises the metrics we use to evaluate tactic pairs. The three selected task performance metrics are well established for the analysis of swarm shepherding systems; see, for example, El-Fiqi et al. (2020), Strömbom et al. (2014), and Debie et al. (2021), who further discuss measures of this class; they are defined below:

1. Mission Success indicates whether the overall goal was achieved.
2. Mission Completion Rate measures the effectiveness of a selected control strategy for a mission, indicating proximity to the goal location from the starting position of the swarm.
3. Mission Speed indicates the effectiveness of a selected control strategy in terms of the average speed of the control agent to move the swarm throughout the mission.

In addition to the three task performance measures discussed above, we present three measures of stability. The two adaptability aspects include: decision changes and separated agents. Exploring the number of decision changes is an established metric, for example, to bound the computations of an agent (Martinez-Gil et al., 2012) or investigate the effect of environmental conditions on decision frequency (Mills et al., 2015). In our setting, we use this metric to quantify the impact a swarm context has on the number of decision changes a control agent needs to make. In addition, quantifying spatially separated agents can help understand collective decision processes and their impacts on a group, for example, the evolution of distinct population groups or agent couplings (Mills et al., 2015). Finally, we use the number of separated agents to measure the impact of the control agent's influence on the stability of the swarm.

The first stability metric is mission decision stability, highlighting whether a small change in the system will result in a large decision space change. We define the function  $\chi_1$  as

$$\chi_1(\sigma(t), \sigma(t+1)) = \begin{cases} 1, & \text{if } \sigma(t) \neq \sigma(t+1) \\ 0, & \text{otherwise} \end{cases}, \quad (6)$$

which acts as a flag for when the control agent switches between behaviours in  $\sigma$ . For example, if the behaviour at  $t$  is  $\sigma_2$  and subsequently at  $t+1$  is  $\sigma_1$ , then  $\chi$  returns the value of 1. If  $\sigma$  does not change for  $t+1$ , then 0 is returned.

The second stability metric is decision swarm stability, which reports the relationship between the number of separated swarm agents and the number of decisions made by the control agent, highlighting whether swarm instability (fracture) is related to control decision changes. We define the function  $\chi_2$  as

$$\chi_2(\pi_i, \Pi^*) = \begin{cases} 1, & \text{if } \pi_i \notin \Pi^* \\ 0, & \text{otherwise} \end{cases}, \quad (7)$$

where  $\Pi^*$  is the cluster size with the largest number of  $\pi$  agents, calculated using the k-means algorithm. We evaluate the position of each  $\pi_i$  to determine whether a swarm agent is separated from this cluster; if the agent is assessed to be separated, we return the value 1 and 0 otherwise.

**Table 7** A comparison between the performance of TP<sub>5</sub> and the tactical pair with the highest mean in each scenario

(a) Metrics M1-M2

	MS		MCR	
	Best TP	TP <sub>5</sub>	Best TP	
$\hat{S}$	<b>0.68 ± 0.47</b> (TP <sub>4</sub> )	0.64 ± 0.48	<b>0.02 ± 0.02</b> (TP <sub>4</sub> )	0.91 ± 2.90
S1	<b>0.90 ± 0.31</b> (TP <sub>4</sub> )	0.73 ± 0.45	<b>0.02 ± 0.02</b> (TP <sub>4</sub> )	0.42 ± 0.80
S2	<b>0.87 ± 0.35</b> (TP <sub>4</sub> )	0.33 ± 0.48*	<b>0.02 ± 0.02</b> (TP <sub>4</sub> )	3.81 ± 7.79*
S3	<b>0.50 ± 0.51</b> (TP <sub>5</sub> )	<b>0.50 ± 0.51</b>	<b>0.02 ± 0.01</b> (TP <sub>4</sub> )	0.81 ± 2.23*
S4	<b>0.90 ± 0.31</b> (TP <sub>9</sub> )	0.62 ± 0.49*	<b>0.02 ± 0.01</b> (TP <sub>4</sub> )	0.97 ± 1.87*
S5	<b>1.00 ± 0.00</b> (TP <sub>5</sub> )	<b>1.00 ± 0.00</b>	<b>0.01 ± 0.00</b> (TP <sub>9</sub> )	0.01 ± 0.01
S6	<b>0.83 ± 0.38</b> (TP <sub>4</sub> )	0.37 ± 0.49*	<b>0.02 ± 0.01</b> (TP <sub>4</sub> )	2.05 ± 4.19*
S7	<b>1.00 ± 0.00</b> (TP <sub>4</sub> )	1.00 ± 0.00	<b>0.01 ± 0.01</b> (TP <sub>21</sub> )	0.01 ± 0.01
S8	<b>0.30 ± 0.47</b> (TP <sub>4</sub> )	0.07 ± 0.25*	<b>0.02 ± 0.01</b> (TP <sub>4</sub> )	1.81 ± 0.77*
S9	<b>0.81 ± 0.40</b> (TP <sub>5</sub> )	<b>0.81 ± 0.40</b>	<b>0.03 ± 0.01</b> (TP <sub>4</sub> )	0.18 ± 0.54*
S10	<b>1.00 ± 0.00</b> (TP <sub>4</sub> )	0.37 ± 0.49*	<b>0.01 ± 0.01</b> (TP <sub>24</sub> )	0.66 ± 0.60*
S11	<b>0.97 ± 0.18</b> (TP <sub>25</sub> )	0.95 ± 0.23	<b>0.01 ± 0.01</b> (TP <sub>25</sub> )	0.06 ± 0.19

(b) Metrics M3-M4

	MSP		MDS	
	Best TP	TP <sub>5</sub>	Best TP	
$\hat{S}$	<b>0.16 ± 0.12</b> (TP <sub>4</sub> )	0.09 ± 0.08	<b>0.27 ± 0.23</b> (TP <sub>4</sub> )	0.13 ± 0.19
S1	<b>0.23 ± 0.12</b> (TP <sub>4</sub> )	0.08 ± 0.06	<b>0.37 ± 0.18</b> (TP <sub>4</sub> )	0.07 ± 0.13
S2	<b>0.12 ± 0.07</b> (TP <sub>4</sub> )	0.04 ± 0.05*	<b>0.33 ± 0.19</b> (TP <sub>4</sub> )	0.10 ± 0.19*
S3	<b>0.05 ± 0.04</b> (TP <sub>5</sub> )	<b>0.05 ± 0.04</b>	<b>0.30 ± 0.47</b> (TP <sub>15</sub> )	0.09 ± 0.13
S4	<b>0.21 ± 0.10</b> (TP <sub>9</sub> )	0.07 ± 0.08*	<b>0.38 ± 0.19</b> (TP <sub>4</sub> )	0.06 ± 0.12*
S5	<b>0.26 ± 0.08</b> (TP <sub>9</sub> )	0.17 ± 0.07*	<b>0.32 ± 0.23</b> (TP <sub>4</sub> )	0.06 ± 0.05*
S6	<b>0.23 ± 0.14</b> (TP <sub>4</sub> )	0.23 ± 0.14	<b>0.32 ± 0.19</b> (TP <sub>4</sub> )	0.04 ± 0.11*
S7	<b>0.23 ± 0.04</b> (TP <sub>4</sub> )	0.19 ± 0.04	<b>0.97 ± 0.18</b> (TP <sub>15</sub> )	0.50 ± 0.00
S8	<b>0.04 ± 0.02</b> (TP <sub>24</sub> )	0.04 ± 0.04*	<b>0.12 ± 0.20</b> (TP <sub>4</sub> )	0.02 ± 0.06*
S9	<b>0.05 ± 0.02</b> (TP <sub>5</sub> )	<b>0.05 ± 0.02</b>	<b>0.43 ± 0.50</b> (TP <sub>11</sub> )	0.11 ± 0.12*
S10	<b>0.24 ± 0.10</b> (TP <sub>24</sub> )	0.06 ± 0.06*	<b>0.40 ± 0.14</b> (TP <sub>4</sub> )	0.02 ± 0.05*
S11	<b>0.21 ± 0.10</b> (TP <sub>24</sub> )	0.15 ± 0.08	<b>0.60 ± 0.50</b> (TP <sub>13</sub> )	0.16 ± 0.21*

(c) Metrics M4-M6

	DSS		MSS	
	Best TP	TP <sub>5</sub>	Best TP	
$\hat{S}$	<b>0.29 ± 0.87</b> (TP <sub>25</sub> )	1.14 ± 1.81	<b>0.12 ± 0.10</b> (TP <sub>5</sub> )	<b>0.12 ± 0.10</b>
S1	<b>0.18 ± 0.26</b> (TP <sub>25</sub> )	0.83 ± 1.39	<b>0.15 ± 0.10</b> (TP <sub>5</sub> )	<b>0.15 ± 0.10</b>
S2	<b>0.03 ± 0.02</b> (TP <sub>25</sub> )	1.65 ± 2.20	<b>0.14 ± 0.09</b> (TP <sub>4</sub> )	0.06 ± 0.10*
S3	<b>0.06 ± 0.05</b> (TP <sub>19</sub> )	1.74 ± 2.42*	<b>0.07 ± 0.07</b> (TP <sub>5</sub> )	<b>0.07 ± 0.07</b>
S4	<b>0.10 ± 0.14</b> (TP <sub>25</sub> )	1.02 ± 1.81	<b>0.14 ± 0.07</b> (TP <sub>9</sub> )	0.13 ± 0.11*
S5	<b>0.14 ± 0.11</b> (TP <sub>25</sub> )	0.41 ± 0.42	<b>0.18 ± 0.06</b> (TP <sub>1</sub> )	0.16 ± 0.03
S6	<b>0.06 ± 0.06</b> (TP <sub>25</sub> )	0.45 ± 0.86	<b>0.08 ± 0.04</b> (TP <sub>4</sub> )	0.05 ± 0.07*

**Table 7** (continued)

(c) Metrics M4-M6

	DSS		MSS	
	Best TP	TP <sub>5</sub>	Best TP	
S7	<b>0.41 ± 0.29</b> (TP <sub>16</sub> )	2.03 ± 0.43	<b>0.32 ± 0.06</b> (TP <sub>18</sub> )	0.26 ± 0.06
S8	<b>0.03 ± 0.02</b> (TP <sub>10</sub> )	0.25 ± 0.72	<b>0.06 ± 0.12</b> (TP <sub>24</sub> )	0.01 ± 0.02*
S9	<b>0.06 ± 0.04</b> (TP <sub>19</sub> )	2.21 ± 3.44*	<b>0.11 ± 0.06</b> (TP <sub>5</sub> )	<b>0.11 ± 0.06</b>
S10	<b>0.11 ± 0.08</b> (TP <sub>25</sub> )	0.23 ± 0.37*	<b>0.15 ± 0.04</b> (TP <sub>19</sub> )	0.06 ± 0.08*
S11	<b>0.30 ± 0.28</b> (TP <sub>25</sub> )	1.17 ± 1.69	<b>0.18 ± 0.07</b> (TP <sub>5</sub> )	<b>0.18 ± 0.07</b>

The best performance is highlighted in a boldface font and statistical significance using an asterisk

The final stability metric is mission swarm stability, indicating whether the swarm control agent can overcome instability in the swarm, such as  $\pi_i$  agents with considerable repulsion strengths for  $W_{\pi\pi}$  or  $W_{\pi\beta}$ .

We evaluate and compare the performance of TPs in two ways. The first uses TP<sub>5</sub> as a baseline because it resembles the algorithm by Strömbom et al. (2014). The best TP with the highest performance in each scenario is then chosen and is labelled as *Best TP*. The best TP is determined for each metric and scenario by calculating the mean and selecting the TP with the highest mean for the particular setting. We then conduct a *t* test to assess whether the difference between the results of TP<sub>5</sub> and the best TP is statistically significant. Table 7 compares performance on each metric and scenario to that of TP<sub>5</sub>. It can be seen that TP<sub>5</sub> is outperformed by another TP across most scenarios and metrics. This demonstrates that using a single TP for homogeneous and heterogeneous swarms parameterised per that of Strömbom et al. (2014) may not be suitable in all circumstances for swarm control, invariant of the metric used to compare performance. In fact, in the 66 cases shown in the table, TP<sub>5</sub> was the best on nine occasions, and only on one of the six metrics in terms of average performance.

In this section, we have systematically explored the parameterisation of autonomous behaviours (atomic collect and drive actions) for a shepherding agent to control a swarm in both homogeneous and heterogeneous settings. We have uncovered that a rule-based approach with a fixed tactic pair in all scenarios is not optimal when scenarios deviate from the classic assumption of homogeneity; that is, the swarm becomes heterogeneous. However, through systematic variations of behaviour combinations, we have demonstrated that mission completion remains high when variations of autonomous behaviours are used, an invariant of the metric to assess performance. The upshot is that a slight variation in the parameterisation of the atomic collect or drive behaviour is sufficient to minimise performance degradation across these scenarios. This quantifies that it can effectively and efficiently control swarms with disparate agent type distributions in shepherding contexts. The remaining challenge for a control agent is to identify the swarm's situation so that it can deliver meaningful real-time guidance from recognition (Bakar et al., 2016). In the following section, we present an intelligent control agent. The intelligent control agent recognises the swarm context and uses this information to select an appropriate TP response.



**Table 8** Summary of experiment design and evaluation measures to quantify the effect of a context-awareness system for swarm control

No.	Hypothesis	Metrics
1	A context-aware system will increase the shepherding agent mission performance	MS and mission length (Fujioka 2017; Himo et al. 2022; Strömbom et al. 2014), and number of $x_i$ influenced by $\beta$ (Strömbom et al. 2014)
2	A context-aware system will increase the shepherding agent control effectiveness for herding tasks	Number of separated swarm agents (Goel et al. 2019), MDS, DSS, and MSS
3	A context-aware system will increase the shepherding agent herding efficiency	$\Pi$ and $\beta$ distance travelled (Fujioka 2017), MSP, and MCR

**Table 9** Mission effectiveness analysis of a swarm control agent with and without the presented context-awareness system

Metrics	Mission success rate		Mission length (all trials)		Mission length (successful trials)		Num. of $\pi_i$ influenced by $\beta$	
	With	Without	With	Without	With	Without	With	Without
	$\hat{S}$	<b>74% ± 28%</b>	64% ± 33%	<b>2,157.50 ± 1,959.90</b>	3,320.10 ± 1,646.60*	<b>1,086.50 ± 909.98</b>	2,275.10 ± 1,111.80*	<b>7.17 ± 8.52</b>
S1	<b>90%</b>	70%	<b>1,136.40 ± 1,414.50</b>	2,956.30 ± 1,635.00*	<b>690.60 ± 121.40</b>	2,016.60 ± 758.80*	<b>13.05 ± 8.61</b>	3.84 ± 5.44*
S2	<b>100%</b>	30%	<b>665.00 ± 0138.00</b>	4,721.40 ± 852.60*	<b>665.00 ± 138.00</b>	3,723.70 ± 1,066.80*	<b>11.31 ± 8.41</b>	2.99 ± 6.19*
S3	<b>100%</b>	90%	<b>863.10 ± 0271.90</b>	2,229.00 ± 1,077.50*	<b>863.10 ± 271.90</b>	1,904.60 ± 349.00*	<b>8.87 ± 8.27</b>	4.22 ± 6.66*
S4	<b>50%</b>	30%	<b>3,702.60 ± 1,674.30</b>	4,393.40 ± 1,496.90	<b>2,256.20 ± 1,037.70</b>	2,630.30 ± 1,850.10	<b>5.43 ± 7.9</b>	3.32 ± 6.24*
S5	40%	<b>50%</b>	<b>4,099.00 ± 1,391.90</b>	4,507.90 ± 741.20	<b>2,524.00 ± 547.20</b>	3,866.80 ± 456.80*	<b>3.79 ± 6.67</b>	3.48 ± 5.33*
S6	80%	<b>90%</b>	<b>1,617.60 ± 1,866.70</b>	2,571.30 ± 1,091.30*	<b>734.80 ± 161.60</b>	2,284.90 ± 645.70*	<b>9.41 ± 9.32</b>	4.08 ± 6.6*
S7	90%	<b>100%</b>	<b>989.00 ± 1,462.60</b>	1,891.60 ± 1,022.30*	<b>526.80 ± 53.90</b>	1,891.60 ± 1,022.30*	<b>12.87 ± 7.31</b>	6.21 ± 7.94*
S8	<b>90%</b>	80%	<b>1,187.50 ± 1,407.10</b>	2,447.20 ± 1,490.40*	<b>747.30 ± 218.80</b>	1,771.70 ± 499.00*	<b>11.65 ± 9.05</b>	4.35 ± 5.91*
S9	100%	100%	<b>768.70 ± 73.60</b>	1,182.80 ± 202.70*	<b>768.70 ± 73.60</b>	1,182.80 ± 202.70*	<b>14.15 ± 8.05</b>	9.17 ± 9.06*
S10	<b>50%</b>	0%	<b>3,879.20 ± 1,696.20</b>	5,149.00 ± 0*	<b>2,609.40 ± 1,562.90</b>	N/A*	<b>8.8 ± 8.97</b>	3.88 ± 6.79*
S11	20%	<b>60%</b>	4,824.90 ± 796.30	<b>4,471.6 ± 696.30</b>	<b>3,528.50 ± 1,226.80</b>	4,020.00 ± 510.80	<b>3.66 ± 6.14</b>	3.02 ± 5*

Bold values indicate the best performance

A star indicates that the difference is statistically significant

## 5 AI2A: actionable-information to actuators

In this section, we take the findings a step further by integrating the metrics into the decision-making process of the control agent. We use three categories of measures of effectiveness and efficiency for the control agent. These categories extend the metrics we present in Table 6 as discussed below.

The two measures of effectiveness categories are for mission and control, whilst the measures of efficiency category are for herding.

Mission effectiveness assesses the performance of the context-aware agent to reach the goal. Control effectiveness measures the impact of the context-aware control agent's tactic pair selection and behaviour modulation on the swarm and mission. Herding efficiency explores the impact of the context-aware system on the physical aspects of the system, such as distance travelled and speed of movement. Each of our lenses explores 11 scenarios (four heterogeneous, S1–S4, and seven homogeneous, S5–S11) for evaluation, comparing a reactive control agent with and without augmentation of the context-awareness system; 110 trials were completed for each setting (with and without context-awareness), yielding 220 trials across 11 scenarios with swarm size,  $N = 20$ . Each scenario-pair setting (with and without context-awareness) is parameterised identically, such that both settings start with a classic reactive control agent ( $\sigma_1^{D100}$ ,  $\sigma_2^{F2G}$ ,  $\sigma^{c_2} = 1$  and  $\sigma^{c_3} = 1$ ) per Strömbom et al. (2014). For trials with the context-awareness system present, the agent is parameterised after the initial observation period of  $\sigma^c$ . Table 8 summarises three hypotheses and the corresponding category of measures used to assess each hypothesis.

### 5.1 Mission effectiveness

Our mission effectiveness analysis considers four aspects; these results are depicted in Table 9. The first is mission success rate, reported as the percentage of trials where  $||P_{\Pi} - P_G|| < R_G$  given  $t \leq T_{\max}$ . The second is the mission length across all trials (successes and failures); as with previous studies, this is the total number of simulation steps. The third measure is mission length across the sub-set of trials for the cases where mission success was achieved. The fourth and final measures of mission effectiveness are the number of  $\pi_i$  agents influenced by  $\beta$ ; Goel et al. (2019) note that “A precise measure of influence using leaders or predators or a combination of leaders and predators to achieve the mission is not adequately studied” (p. 1). We use the interaction radius of  $\beta$  to determine whether a  $\pi_i$  is directly influenced or not; the influence range parameterisation is consistent with that described in Strömbom et al. (2014).

We observe an overall improvement in mission success of 10% for the control agent with the context-aware system; most notably, the best mission success is achieved with heterogeneous swarm scenarios (S1–S4), outperforming the agent without the context-awareness system for all heterogeneous settings. In addition, the context-aware agent observes a decrease in mission length for both all trials and only successful trials, with statistically significant improvement in 72.7% of all trials and 81.8% of successful trials. Finally, for our final mission effectiveness measure (number of  $\pi$  influenced by  $\beta$ ), we observe statistical significance in 100% of trials, indicating that the context-aware control agent influences a more significant number of swarm agents across all homogeneous and heterogeneous scenarios.

**Table 10** Stability measures of a swarm control agent with and without the presented context-awareness system

Metrics	Mean sep. agents				MDS		DSS		MSS	
	With		Without		With	Without	With	Without	With	Without
	With	Without	With	Without	With	Without	With	Without	With	Without
$\hat{S}$	<b>7.18 ± 2.21</b>	7.70 ± 2.26	<b>0.29 ± 0.29</b>	0.24 ± 0.24	<b>1.66 ± 1.30</b>	2.30 ± 2.48	<b>0.10 ± 0.05</b>	0.09 ± 0.06	<b>0.10 ± 0.05</b>	0.09 ± 0.06
S1	8.58 ± 2.59*	<b>6.73 ± 2.40</b>	<b>0.38 ± 0.43</b>	0.15 ± 0.32*	2.12 ± 1.86	<b>1.69 ± 3.00</b>	<b>0.10 ± 0.03</b>	0.10 ± 0.08	<b>0.10 ± 0.03</b>	0.10 ± 0.08
S2	9.49 ± 0.76*	<b>4.81 ± 1.15</b>	<b>0.14 ± 0.08</b>	0.01 ± 0.02*	1.13 ± 0.57*	<b>0.18 ± 0.11</b>	0.11 ± 0.01	<b>0.06 ± 0.10</b>	0.11 ± 0.01	<b>0.06 ± 0.10</b>
S3	7.32 ± 1.76	<b>5.89 ± 2.20</b>	<b>0.55 ± 0.39</b>	0.26 ± 0.40*	2.37 ± 1.42*	<b>1.04 ± 1.26</b>	<b>0.14 ± 0.03</b>	0.15 ± 0.07	<b>0.14 ± 0.03</b>	0.15 ± 0.07
S4	<b>4.56 ± 2.56</b>	8.43 ± 2.60*	0.04 ± 0.08	<b>0.21 ± 0.37</b>	<b>0.34 ± 0.41</b>	4.30 ± 4.76*	0.08 ± 0.09	<b>0.03 ± 0.05</b>	0.08 ± 0.09	<b>0.03 ± 0.05</b>
S5	<b>6.01 ± 2.04</b>	11.97 ± 3.33*	0.09 ± 0.12	<b>0.48 ± 0.49</b>	<b>1.15 ± 1.08</b>	7.16 ± 6.02*	0.05 ± 0.07	<b>0.05 ± 0.05</b>	0.05 ± 0.07	<b>0.05 ± 0.05</b>
S6	7.63 ± 3.36	<b>6.63 ± 2.47</b>	<b>0.23 ± 0.29</b>	0.19 ± 0.31	1.51 ± 1.38	<b>1.16 ± 1.91</b>	0.09 ± 0.05*	<b>0.14 ± 0.06</b>	0.09 ± 0.05*	<b>0.14 ± 0.06</b>
S7	8.63 ± 2.75*	<b>7.18 ± 1.66</b>	<b>0.23 ± 0.28</b>	0.09 ± 0.07	1.60 ± 1.42*	<b>0.50 ± 0.33</b>	0.10 ± 0.04*	<b>0.15 ± 0.06</b>	0.10 ± 0.04*	<b>0.15 ± 0.06</b>
S8	9.60 ± 3.05	<b>8.18 ± 2.86</b>	<b>0.51 ± 0.43</b>	0.08 ± 0.10*	3.11 ± 2.12*	<b>0.74 ± 0.92</b>	<b>0.09 ± 0.03</b>	0.09 ± 0.05	<b>0.09 ± 0.03</b>	0.09 ± 0.05
S9	5.69 ± 0.82	<b>5.20 ± 0.96</b>	0.65 ± 0.38	<b>0.92 ± 0.25</b>	<b>2.07 ± 0.88</b>	2.45 ± 0.75	<b>0.18 ± 0.03</b>	0.20 ± 0.04	<b>0.18 ± 0.03</b>	0.20 ± 0.04
S10	<b>5.88 ± 2.37</b>	9.06 ± 2.99*	<b>0.29 ± 0.40</b>	0 ± 0*	<b>1.78 ± 1.64</b>	2.11 ± 3.42	<b>0.08 ± 0.09</b>	0 ± 0*	<b>0.08 ± 0.09</b>	0 ± 0*
S11	<b>5.53 ± 2.28</b>	10.59 ± 2.26*	0.10 ± 0.32*	<b>0.22 ± 0.33</b>	<b>1.04 ± 1.52</b>	3.96 ± 4.77*	<b>0.03 ± 0.08</b>	0.06 ± 0.06	<b>0.03 ± 0.08</b>	0.06 ± 0.06

Bold values indicate the best performance

**Table 11** Efficiency analysis of a swarm control agent with and without the presented context-awareness system

Metrics	Swarm total dist.		Agent total dist.		MSP		MCR	
	With	Without	With	Without	With	Without	With	Without
	$\hat{S}$	171.60 ± 11.20*	<b>99.58 ± 34.70</b>	226.52 ± 29.41	227.24 ± 62.60	<b>0.18 ± 0.60</b>	0.07 ± 0.03*	<b>0.03 ± 0.03</b>
S1	175.39 ± 10.67*	<b>97.38 ± 37.11</b>	<b>235.13 ± 16.63</b>	248.03 ± 38.20	<b>0.24 ± 0.09</b>	0.07 ± 0.04*	<b>0.01 ± 0.01</b>	0.48 ± 0.81
S2	173.59 ± 10.07*	<b>133.07 ± 21.56</b>	<b>227.23 ± 10.69</b>	248.59 ± 83.37*	<b>0.27 ± 0.04</b>	0.03 ± 0.01*	<b>0.01 ± 0</b>	1.08 ± 0.96*
S3	173.59 ± 10.08*	<b>88.30 ± 25.70</b>	<b>225.32 ± 15.81</b>	239.09 ± 35.45	<b>0.23 ± 0.09</b>	0.09 ± 0.03*	<b>0.01 ± 0.01</b>	0.07 ± 0.19
S4	168.82 ± 15.45*	<b>67.33 ± 53.43</b>	211.89 ± 36.72	<b>196.19 ± 77.75</b>	<b>0.06 ± 0.05</b>	0.04 ± 0.05*	<b>0.04 ± 0.06</b>	3.45 ± 4.39*
S5	159.24 ± 11.10*	<b>56.20 ± 51.69</b>	235.49 ± 37.04	<b>216.65 ± 88.45</b>	<b>0.05 ± 0.02</b>	0.03 ± 0.02	<b>0.07 ± 0.07</b>	2.08 ± 3.48
S6	172.37 ± 10.12*	<b>103.96 ± 53.97</b>	<b>226.91 ± 31.10</b>	247.91 ± 34.92	<b>0.21 ± 0.11</b>	0.08 ± 0.03*	<b>0.02 ± 0.01</b>	0.05 ± 0.13*
S7	175.44 ± 10.39*	<b>141.17 ± 9.35</b>	222.63 ± 22.57	<b>212.48 ± 74.71</b>	<b>0.31 ± 0.10</b>	0.11 ± 0.04*	0.02 ± 0.01*	<b>0.01 ± 0</b>
S8	174.80 ± 11.13*	<b>125.69 ± 25.37</b>	235.28 ± 15.14	<b>208.97 ± 74.12</b>	<b>0.23 ± 0.09</b>	0.09 ± 0.05*	<b>0.02 ± 0.01</b>	0.28 ± 0.61
S9	176.36 ± 10.05*	<b>60.02 ± 9.89</b>	<b>221.15 ± 7.50</b>	225.30 ± 6.58	<b>0.23 ± 0.01</b>	0.15 ± 0.02*	<b>0.01 ± 0.01</b>	0.02 ± 0.01
S10	173.04 ± 9.26*	<b>110.95 ± 49.38</b>	221.67 ± 37.74	<b>190.38 ± 07.93</b>	<b>0.06 ± 0.04</b>	0.02 ± 0.01*	<b>0.04 ± 0.04</b>	2.64 ± 2.31*
S11	164.92 ± 14.84*	<b>111.23 ± 43.51</b>	<b>229.00 ± 92.60</b>	266.00 ± 67.13	0.04 ± 0.01	0.04 ± 0.01	<b>0.07 ± 0.06</b>	0.55 ± 0.91

Bold values indicate the best performance

## 5.2 Control effectiveness

We consider four metrics to analyse control effectiveness, including three metrics as presented in Table 6; these results are depicted in Table 10. The first metric is the mean number of separated  $\pi_i$  agents from the central swarm cluster  $\Pi$  over each trial. We observe an overall lower mean number of separated agents for the control agent with context-awareness. Mission decision stability effectively measures the sensitivity of the mission outcome to the number of behaviour changes, manifesting as the count of  $\sigma_1 \rightarrow \sigma_2$  and  $\sigma_2 \rightarrow \sigma_1$  changes. We observe significance in favour of the control agent with the context-awareness system in 45% of scenarios, achieving the best outcome in 64% of all scenarios.

Decision swarm stability is the ratio between the number of separated  $\pi_i$  and the number of decision changes  $\beta$ , averaged over the overall mission. The control agent without context-awareness performs marginally better in 55% of scenarios, although the significance is only recorded in 27%. Of interest, the control agent with context-awareness has better aggregate performance across all scenarios; however, the results are not significant. The final metric is mission swarm stability, measuring the sensitivity of mission success to the number of separated swarm agents. As with decision swarm stability, we observe a marginal distinction between the control agent with and without context-awareness. In this case, the context-aware-enabled agent is best in 55% of scenarios, with significance recorded in 18%.

## 5.3 Herding efficiency

Our final perspective is herding efficiency, which captures the physical movements of the agents and the herd in the system; results are depicted in Table 11. The first metric is the total swarm distance moved, measuring the cumulative path distance from  $P_{\Pi}^{t=0}$  to  $P_{\Pi}^{t=T}$ . We calculate  $P_{\Pi}^t = 1/N \sum_{i=1}^N P_{\pi_i}^t$ . We observe statistical significance across all scenarios, with the control agent without the context-aware system minimising the total distance travelled by the swarm. Comparing the settings with and without context-awareness, we note that the context-aware system results have a significantly lower deviation in the distance travelled per scenario ( $Z = 2.63, p < 0.01$ ). This is an important finding in our work concerning system energy utilisation. In a physical robotic system, it is crucial to ensure that energy use is predictable, enabling performance guarantees to be placed on the system. Whilst a control agent with our context-awareness system increases the distance that the swarm travels, the empirical variance is stable between scenarios. The control agent without context-awareness observes a standard deviation that approximately equals the magnitude of the mean distance travelled (see S5 for example), with a lower bound approaching that observed for the control agent with context-awareness. The increased total swarm distance travelled of the context-aware system relates to the objective of this control agent to maintain a cohesive swarm. In the classic model, the agent is only concerned with the relative position of the centre of mass for the swarm and the goal. The second factor of further consideration here is regarding the control agent type implementation, particularly that for the control agent without context-awareness as defined conditions for  $\sigma_2$  must be met prior to the execution of  $\sigma_1$ .

Our second metric is the control agent's total distance travelled, calculated as per the swarm total distance travelled, however utilising only  $P_{\beta}$ . For this metric, we see the statistical significance in only one scenario; for all other settings, the total distance each agent

type moves is approximately the same. Our third metric is mission speed, in which we observe significance in over 90% of scenarios, with the context-aware-enabled agent demonstrating higher speed over different scenarios. Our final metric is the mission completion rate. This metric provides insight into the efficiency of the control strategy, mainly when the mission is unsuccessful. The control agent with a context-awareness system achieves the best performance in 90% of scenarios, although the significance is recorded in only 36%.

## 5.4 Discussion

We focus on synthesising results across the three lenses of mission effectiveness, control effectiveness, and herding efficiency, drawing insight from the metrics discussed earlier in this section. Our first discussion considers the relationship between mission length and the total swarm distance moved, considering each metric independently provides an understanding of the lenses we discussed previously. However, when compared directly, we observe where the shorter mission length is associated with further total distances moved, particularly where the control agent is augmented with the context-aware system. We found a strong negative correlation between mission length (all trials) and the swarm total distance moved for the control agent with context-awareness ( $r(9) = -0.8, p < 0.01$ ) and no significant correlation for the control agent without context-awareness ( $r(9) = 0.02, p > 0.05$ ). Similar results are observed for mission length (successful trials) and the total swarm distance moved. We found a strong negative correlation for the control agent with context-awareness ( $r(9) = -0.77, p < 0.01$ ) and no significant correlation for the control agent without context-awareness ( $r(9) = -0.05, p > 0.05$ ). The statistically significant correlation for the control agent with context-awareness is unexpected as previous studies report that the likelihood of success decreases proportionally with the mission length (Strömbom et al., 2014), suggesting that the longer the mission continues, the lower the success rate will be. Further investigation is required to understand the nature of the relationship between the swarm's total distance moved and the mission length.

One possible way to consider this outcome is to develop insight into this phenomenon when considering the distinct mission speeds. For the control agent with context-awareness, there exists a strong negative correlation between mission length (all trials) and mission speed ( $r(9) = -0.97, p < 0.001$ ) and between mission length (successful trials) and mission speed ( $r(9) = -0.95, p < 0.001$ ), as well as a solid positive correlation between swarm total distance moved and mission speed ( $r(9) = 0.78, p < 0.01$ ). On the other hand, for the control agent without a context-aware system, these results hold in the case of all trials ( $r(9) = -0.98, p < 0.001$ ) are weak negative correlated for successful trials ( $r(9) = -0.39, p > 0.05$ ), although no significant correlation exists between the total swarm distance moved and mission speed ( $r(9) = -0.06, p > 0.05$ ). The strong negative correlation between mission speed and mission length metrics is expected for the control agent with context-awareness. This intuitively can be interpreted that the faster an agent moves the swarm, the less time the mission will take.

We hypothesise that the difference between the control agent with and without context-awareness is due, in part, to frequency behaviour modulation. For the agent without context-awareness, a *stall distance* exists when a control agent becomes too close to any swarm agent (Perry et al., 2021).

We introduce behaviour parameterisations to address the stall distance gap by maintaining the selected action modulation point invariant any  $\pi_i$  response. As the control agent

without context-awareness does not have this capability and will stall when within a certain distance to a swarm agent, behaviour parameterisations could account for the incongruent observations between mission length and total swarm distance moved.

Our second point of discussion centres on the mean number of separated  $\pi_i$  from the giant  $\Pi$  cluster. Our initial expectation was that the control agent with context-awareness would result in a statistically significantly lower (i.e. better) mean number of separated swarm agent as a function of the number of decision changes the control agent makes. Recall that in the context-aware setting, the upper bound of this rate is modulated. Correlation analysis reveals a weak, non-significant correlation between these features for either the control agent with ( $r(9) = 0.30, p > 0.05$ ) or without ( $r(9) = 0.30, p > 0.05$ ) the context-awareness system. We postulate that the introduction of  $\mathcal{L}$  in Eq. 3 may contribute to an increase in separated swarm agent as we allow drive actions to occur prior to the behaviour transition threshold. What this means is that for the agent with the context-aware system, we change the decision boundary to allow a drive action to occur *prior* to the boundary condition being met for all members of the swarm. Whilst this does result in a greater number of separated swarm members, the overall mission length (all trials and successful trials) is significantly reduced. This outcome is expected as separated swarm agents may become isolated more easily for values of  $\mathcal{L} \leq 0.75$  used in this study; investigation of systems with obstacles present is likely to provide further interesting results.

## 6 Conclusions and future work

Swarm control can be difficult due to the requirement to understand, control, and anticipate the expected responses of swarm members. Typically, environmental complexities with static decision models are used to evaluate swarm control model performance. These explorations often maintain the standard baseline model to evaluate its performance through a particular lens, such as where this model breaks down. We assert that environmental influences that still impact the swarm agents continue to manifest as recognition and control problems for the swarm agent. By focussing on the individual agents within the swarm, their characteristics become the situations we identify. This enables us to determine the impact of the characteristic changes at individual or collective levels without specifying the properties of the surrounding environment.

In this paper, we have introduced a context-awareness system for a reactive shepherding control agent, demonstrating significant improvements in mission effectiveness. Our context-awareness system is an information marker-based approach that focusses on the structuring and organisation of information to understand disparate contexts and situations of a swarm. The context-awareness system is a decision algorithm that focusses the attention of the swarm control agent on particular aspects of the swarm, reducing the search space of possible behaviours in response to the actions of the swarm. To enable this system, we conduct a systematic behaviour study that investigates the applicability of disparate control actions across distinct homogeneous and heterogeneous scenarios.

There are several avenues of future work to refine our context-awareness system whilst building on the architecture presented. The first is related to the selection of the decision model. The implemented context-awareness decision model is a rule-based reasoning engine. Further research is required to select and evaluate alternative decision models suitable for the deployment of robotic platforms, particularly in settings where sensor noise will perturb normal system operations. The second avenue of future work is to study the



impact of incorrect context. In this work, we used a closed system with a static environment, defined agent types, and declared goal location. In real-world settings, this information may not always be available to an AI control agent, for instance, where a new type of agent in the swarm is present. In a shepherding setting, this could be the evaluation of heterogeneous flocks with agents not previously observed. A third avenue is to investigate the development of distributions of each metric for the context-aware control agent. This could provide the context-awareness system with the ability to predict, in advance, the likelihood of success. The final avenue of future work is to consider adaptation and learning in the swarm and its effect on the control agent's strategy. It is well established in biological settings that cognitive agents adapt and learn over time. However, these impacts remain open questions for swarm research.

**Acknowledgements** The authors wish to thank our research colleagues Kate J. Yaxley and Daniel P. Baxter for providing constructive feedback about the manuscript.

**Author contributions** AH conceptualised the study, designed and performed the experimentation (including model development, code implementation, and simulation output analysis), and prepared the manuscript. AH, DR, and HA supervised the study design, experimental conduct, and reviewed and commented on the manuscript at all stages.

**Funding** Open Access funding enabled and organized by CAUL and its Member Institutions. The authors received no financial support for the research, authorship, and/or publication of this article.

**Data availability statement** The code that generated the data for this paper can be accessed at <https://github.com/Fadamos/IntelligentControlAgent>.

## Declarations

**Conflict of interest** All authors declared that they have no conflicts for the research, authorship, and/or publication of this article.

**Ethics approval and consent to participate** Not applicable.

**Consent for publication** Not applicable.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Abbass, H., Harvey, J. & Yaxley, K. (2018). Lifelong testing of smart autonomous systems by shepherding a swarm of watchdog artificial intelligence agents.
- Abpeikar, S., Kasmarik, K., Tran, P., & Garratt, M. (2022). *Transfer learning for autonomous recognition of Swarm behaviour in UGVs. AI 2021: Advances in artificial intelligence. AI 2022* (Vol. 13151). Cham: Springer.
- Auletta, F., Fiore, D., Richardson, M. J., & di Bernardo, M. (2022). Herding stochastic autonomous agents via local control rules and online target selection strategies. *Autonomous Robots*, 46, 469–481.

- Bakar, U.A.B.U.A., Ghayvat, H., Hasanm, S.F. & Mukhopadhyay, S.C. (2016). Activity and anomaly detection in smart home: A survey. In *Next generation sensors and systems* (pp. 191–220). Springer International Publishing.
- Bredeche, N., & Fontbonne, N. (2022). Social learning in swarm robotics. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 377(1843), 20200309.
- Chen, S., Pei, H., Lai, Q., & Yan, H. (2019). Multitarget tracking control for coupled heterogeneous inertial agents systems based on flocking behavior. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(12), 2605–2611.
- Cowling, P., & Gmeinwieser, C. (2010). Ai for herding sheep. In *Proceedings of the sixth AAAI conference on artificial intelligence and interactive digital entertainment* (pp. 2–7). AAAI Press.
- Debie, E., Singh, H., Elsayed, S., Perry, A., Hunjet, R. & Abbass, H. (2021). A neuro-evolution approach to shepherding swarm guidance in the face of uncertainty. In *2021 IEEE international conference on systems, man, and cybernetics (SMC)* (pp. 2634–2641).
- El-Fiqi, H., Campbell, B., Elsayed, S., Perry, A., Singh, H. K., Hunjet, R., & Abbass, H. A. (2020). The limits of reactive shepherding approaches for swarm guidance. *IEEE Access*, 8, 214658–214671.
- Fujioka, K. (2017). Comparison of shepherding control behaviors. *Tencon 2017–2017 IEEE region 10 conference* (pp. 2426–2430).
- Go, C.K., Koganti, N., Ikeda, K. (2021). Solving the shepherding problem: Imitation learning can acquire the switching algorithm. In *2021 international joint conference on neural networks (IJCNN)* (pp. 1–7).
- Goel, R., Lewis, J., Goodrich, M. & Sujit, P. (2019). Leader and predator based swarm steering for multiple tasks. In *2019 IEEE international conference on systems, man and cybernetics (SMC)*.
- Hasan, Y., Baxter, J.E.G., Castillo, C.A.S., Delgado, E. & Tapia, L. (2022). Flock navigation by coordinated shepherds via reinforcement learning. In *The 15th international workshop on the algorithmic foundations of robotics (WAFR)* (pp. 1–16).
- Hepworth, A. (2021). Activity recognition for shepherding. H. Abbass & R. Hunjet (Eds.), *Shepherding UxVs for Human-Swarm Teaming* (pp. 131–164). Springer
- Hepworth, A.J., Hussein, A., Reid, D.J. & Abbass, H.A. (2023). Swarm analytics: Designing information markers to characterise swarm systems in shepherding contexts. *Adaptive Behavior*.
- Hepworth, A.J., Yaxley, K.J., Baxter, D.P., Joiner, K.F. & Abbass, H. (2020). Tracking footprints in a swarm: Information-theoretic and spatial centre of influence measures. In *2020 IEEE symposium series on computational intelligence (SSCI)* (pp. 2217–2224).
- Himo, R., Ogura, M., & Wakamiya, N. (2022). Iterative shepherding control for agents with heterogeneous responsibility. *Mathematical Biosciences and Engineering*, 19, 3509–3525.
- Hoshi, H., Iimura, I., Nakayama, S., Moriyama, Y. & Ishibashi, K. (2018). Computer simulation based robustness comparison regarding agents' moving-speeds in two and three-dimensional herding algorithms. In *2018 joint 10th international conference on soft computing and intelligent systems (SCIS) and 19th international symposium on advanced intelligent systems (ISIS)* (pp. 1307–1314).
- Hu, J., Turgut, A. E., Krajenik, T., Lennox, B., & Arvin, F. (2022). Occlusion based coordination protocol design for autonomous robotic shepherding tasks. *IEEE Transactions on Cognitive and Developmental Systems*, 14(1), 126–135.
- Hu, J., Wu, H. & Zhan, R. (2022). Wolf pack intelligence: From biological intelligence to cooperative control for swarm robotics. *Advances in guidance, navigation and control*. In *Proceedings of 2020 international conference on guidance, navigation and control (ICGNC 2020)*, Tianjin, China, October 23–25, 2020 (Vol. 644, pp. 4943–4955). Springer.
- Hussein, A., Petraki, E., Elsayah, S. & Abbass, H. (2022). Autonomous swarm shepherding using curriculum-based reinforcement learning. In Faliszewski, P., Mascardi, V., Pelachaud, C. & Taylor, M. (Eds.) *Proc. of the 21st international conference on autonomous agents and multiagent systems (AAMAS 2022)*. Auckland, New Zealand.
- Jang, I., Shin, H.-S., & Tsourdos, A. (2018). Local information-based control for probabilistic swarm distribution guidance. *Swarm Intelligence*, 12, 327–359.
- Jolles, J. W., King, A. J., & Killen, S. S. (2020). The role of individual heterogeneity in collective animal behaviour. *Trends in Ecology & Evolution*, 35(3), 278–291.
- Kengyel, D., Hamann, H., Zahadat, P., Radspieler, G., Wotawa, F. & Schmickl, T. (2015). Potential of heterogeneity in collective behaviors: A case study on heterogeneous swarms. In *International conference on principles and practice of multi-agent systems (prima)* (pp. 201–217).
- Lee, W., & Kim, D. (2017). Autonomous shepherding behaviors of multiple target steering robots. *Sensors*, 17(12), 2729.

- Li, M., Hu, Z., Liang, J. & Li, S. (2012). Shepherding behaviors with single shep herd in crowd management. In Xiao, T., Zhang, L. & Ma, S. (Eds.) *System simulation and scientific computing* (pp. 415–423). Springer
- Lien, J.-M., Bayazit, O., Sowell, R., Rodriguez, S. & Amato, N. (2004). Shepherding behaviors. Ieee international conference on robotics and automation. In *2004, Proceedings. ICRA '04. 2004* (Vol. 4, pp. 4159–4164).
- Lien, J.-M., Rodriguez, S., Malric, J., & Amato, N. (2005). Shepherding behaviors with multiple shepherds. In *Proceedings of the 2005 IEEE international conference on robotics and automation* (pp. 3402–3407).
- Long, N. K., Sammut, K., Sgarioto, D., Garratt, M., & Abbass, H. A. (2020). A comprehensive review of shepherding as a bio-inspired swarm-robotics guidance approach. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 4(4), 523–537.
- Martinez-Gil, F., Lozano, M. & Fernández, F. (2012). Multi-agent reinforcement learning for simulating pedestrian navigation. Vranx, P., Knudson, M. & Grzes, M. (Eds.) *Adaptive and learning agents* (pp. 54–69). Springer.
- Masehian, E., & Royan, M. (2015). Cooperative control of a multi robot flocking system for simultaneous object collection and shepherding. *Computational intelligence: International joint conference, IJCCI 2012 Barcelona, Spain, October 5–7, 2012 revised selected papers* (pp. 97–114). Springer International Publishing.
- Mills, R., Zahadat, P., Silva, F., Mlikic, D., Mariano, P., Schmickl, T. & Correia, L. (2015). Coordination of collective behaviours in spatially separated agents. In *Ecal 2015: the 13th European conference on artificial life* (pp. 579–586).
- Mohamed, R.E., Elsayed, S., Hunjet, R. & Abbass, H. (2021). A graph-based approach for shepherding swarms with limited sensing range. In *2021 IEEE congress on evolutionary computation (CEC)* (pp. 2315–2322).
- Mohanty, N., Gadde, M.S., Sundaram, S., Sundararajan, N. & Sujit, P.B. (2020). Context-aware deep q-network for decentralized cooperative reconnaissance by a robotic swarm. arXiv.
- Nguyen, H.T., Nguyen, T.D., Garratt, M., Kasmarik, K., Anavatti, S., Barlow, M. & Abbass, H.A. (2019). A deep hierarchical reinforcement learner for aerial shepherding of ground swarms. T. Gedeon, K.W. Wong, & M. Lee (Eds.), *Neural information processing* (pp. 658–669). Springer International Publishing.
- Özdemir, A., Gaucei, M. & Groß, R. (2017). Shepherding with robots that do not compute. In *Ecal 2017, the fourteenth European conference on artificial life, at: Lyon, France* (pp. 332–339).
- Pajares Ferrando, S., & Onaindia, E. (2013). Context-aware multi-agent planning in intelligent environments. *Information Sciences*, 227, 22–42.
- Paranjape, A. A., Chung, S., Kim, K., & Shim, D. H. (2018). Robotic herding of a flock of birds using an unmanned aerial vehicle. *IEEE Transactions on Robotics*, 34(4), 901–915.
- Perry, A. (2021). The influence of stall distance on effective shepherding of a swarm. In H. Abbass & R. Hunjet (Eds.), *Shepherding UxVs for human swarm teaming* (pp. 67–83). Springer.
- Reynolds, C. (1987). Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on computer graphics and interactive techniques* (Vol. 21, pp. 25–34). ACM.
- Rizk, Y., Awad, M., & Tunstel, E. (2019). Cooperative heterogeneous multi-robot systems: A survey. *ACM Computing Surveys*, 52(2), 29.
- Shehory, O., & Sturm, A. (2001). *Evaluation of modeling techniques for agent-based systems* (pp. 624–631). Association for Computing Machinery.
- Shepard, D. (1968). A two-dimensional interpolation function for irregularly spaced data. In *Proceedings of the 1968 23rd ACM national conference* (pp. 517–524). Association for Computing Machinery.
- Singh, H., Campbell, B., Elsayed, S., Perry, A., Hunjet, R. & Abbass, H. (2019). Modulation of force vectors for effective shepherding of a swarm: A bi objective approach. In *2019 IEEE congress on evolutionary computation (CEC)* (pp. 2941–2948).
- Strömbom, D., Mann, R. P., Wilson, A. M., Hailes, S., Morton, A. J., Sumpter, D. J. T., & King, A. J. (2014). Solving the shepherding problem: Heuristics for herding autonomous, interacting agents. *Journal of the Royal Society Interface*, 11(100), 20140719.
- Traboulsi, A., & Barbeau, M. (2019). Recognition of drone formation intentions using supervised machine learning. In *2019 international conference on computational science and computational intelligence (CSCI)* (pp. 408–411).
- Tsunoda, Y., Sueoka, Y., Sato, Y., & Osuka, K. (2018). Analysis of local-camera based shepherding navigation. *Advanced Robotics*, 32(23), 1217–1228.

- Varadharajan, V.S., Dyanatkar, S. & Beltrame, G. (2022). Hierarchical control of smart particle swarms. arXiv.
- Vicsek, T., Czirók, A., Ben-Jacob, E., Cohen, I., & Shochet, O. (1995). Novel type of phase transition in a system of self-driven particles. *Physical Review Letters*, 75, 1226–1229.
- Zhang, Q., Hao, Y., Yang, Z., & Chen, Z. (2016). Adaptive flocking of heterogeneous multi-agents systems with nonlinear dynamics. *Neurocomput.*, 216(C), 72–77.
- Zhang, S., Lei, X., Duan, M., Peng, X. & Pan, J. (2022, 06). Herding a flock using a distributed outmost push strategy with multi-robot system. pre-print (pp. 1–18).
- Zhang, S., & Pan, J. (2022). Collecting a flock with multiple sub-groups by using multi-robot system. *IEEE Robotics and Automation Letters*, 7(3), 6974–6981.
- Zhi, J., & Lien, J.-M. (2021). Learning to herd agents amongst obstacles: Training robust shepherding behaviors using deep reinforcement learning. *IEEE Robotics and Automation Letters*, 6(2), 4163–4168.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.