drones

Article

Robotic Herding of Farm Animals Using a Network of Barking Aerial Drones

Xiaohui Li [1], Hailong Huang [2], Andrey V. Savkin [1,*] and Jian Zhang [1]

[1] School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney 2052, Australia; xiaohui.li@unsw.edu.au (X.L.); jian.zhang4@unsw.edu.au (J.Z.)
[2] Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hong Kong, China; hailong.huang@polyu.edu.hk
* Correspondence: a.savkin@unsw.edu.au

Abstract: This paper proposes a novel robotic animal herding system based on a network of autonomous barking drones. The objective of such a system is to replace traditional herding methods (e.g., dogs) so that a large number (e.g., thousands) of farm animals such as sheep can be quickly collected from a sparse status and then driven to a designated location (e.g., a sheepfold). In this paper, we particularly focus on the motion control of the barking drones. To this end, a computationally efficient sliding mode based control algorithm is developed, which navigates the drones to track the moving boundary of the animals' footprint and enables the drones to avoid collisions with others. Extensive computer simulations, where the dynamics of the animals follow Reynolds' rules, show the effectiveness of the proposed approach.

Keywords: autonomous drones; unmanned aerial vehicles (UAVs); robotic herding; shepherding; precision farming; motion control; swarm guidance

Citation: Li, X.; Huang, H.; Savkin, A.V.; Zhang, J. Robotic Herding of Farm Animals Using a Network of Barking Aerial Drones. *Drones* **2022**, *6*, 29. https://doi.org/10.3390/drones6020029

Academic Editor: Abdessattar Abdelkefi

Received: 9 December 2021
Accepted: 18 January 2022
Published: 19 January 2022

1. Introduction

Robot farming plays a critical role in preventing the food crisis caused by future population growth [1]. The past decades have seen the rapid development of robotic crop farming, such as automated crop monitoring, harvesting, weed control, and so forth [2,3]. Deploying robotic automation to improve crop production yields has actually become very popular among farmers. In contrast, research and implementations of robotic livestock farming have been mostly restricted in the fields of virtual fencing [4], animal monitoring and pasture surveying [5,6]. Such applications can improve livestock production yields to some extent. But animal herding as the vital step of livestock farming has long been the least automated. Sheepdogs that have been used for centuries are still the dominant tools of animal herding around the world, and the research on robotic animal herding is still in its infancy. Two main obstacles of robotic animal herding systems are: (1) lack of practical robot-to-animal interactions and the suitable robotic herding platform; and (2) lack of efficient robotic herding algorithm for abundant animals.

The applications of robots to the actual act of animal herding started from the Robot Sheepdog Project in the 1990s [7,8]. These groundbreaking studies achieved gathering a flock of ducks and manoeuvring them to a specified goal position by wheeled robots. The last three decades have only seen a handful of studies on robotic herding with real animals. Recent implementations of robotic animal herding mainly employ ground robots that drive animals through bright colours [9] or collisions [10–12]. Reference [9] shows that the robot initially repulsed the sheep at a distance of 60 m; however, after only two further trials, the repulsion distance drops to 10 m. Besides, such ground legged or wheeled robots might not be agile enough to deal with various terrains during herding. Furthermore, the employed Rover robot in [10], Spot robot in [11] and Swagbot in [12] all cost hundreds of thousands US dollars each and are still in the prototype stages. High prices limit their popularity. Interestingly, the real sheepdogs are also expensive because of their years-long training period. A fully trained sheepdog can cost tens of thousands of US dollars [13]. The most crucial drawback of sheepdogs is that they cannot get rid of biological limitations, that is, ageing and illness.

Besides the platforms, efficient algorithms are also critical to the study of robotic animal herding. Despite the stagnant progress of the herding platforms, research of the related algorithms has experienced considerable development. The prime cause of this contradiction is the recent rapid development of the study on swarm robotics and swarm intelligence [14]. Bio-inspired swarming-based control algorithms for herding swarm robots are receiving much attention in robotics due to the effectiveness of solutions found in nature (e.g., interactions between sheep and dogs). Such algorithms can also be applied to herd flocks of animals. A considerable amount of literature has been published on this topic. For example, paper [15] designs a simple heuristic algorithm for a single shepherd to solve the shepherding problem, based on adaptive switching between collecting the agents when they are too dispersed and driving them once they are aggregated. One unique contribution of [15] is that it conducts field tests with a flock of real sheep and reproduces key features of empirical data collected from sheep—dog interactions. To elaborate the results in [15], reference [16] tests the effects of the step size per unit time of the shepherd and swarm agents, and clarifies that the herding algorithm in [15] is mostly robust regarding swarm agents' moving speeds. Its further study [17] extends the shepherd and swarm agents' motion and influential force vectors to the third dimension.

References [18,19] propose the multi-shepherd control strategies for guiding swarm agents in 2D and 3D environments based on a single continuous control law. The implementation of such strategies requires more shepherds than swarm agents, thus cannot deal with tasks with abundant agents. The level of modulation of the force vector exerted by the shepherd on the swarm agents plays a critical role in herding task success and energy used. Paper [20] designs a force modulation function for the shepherd agent and adopts a genetic algorithm to optimise the energy used by the agent subject to a threshold of success rate. These algorithms and most of the studies in robotic herding, however, have only been carried out in the tasks with tens of swarm agents. The algorithm for efficiently herding abundant swarm agents has not been investigated.

Comparing with ground robots, autonomous drones have superior manoeuvrability and are finding increasing use in different areas, including agriculture [21,22], surveillance [23,24], communications [25] and disaster relief [26]. Particularly, references [21,22] demonstrate the feasibility of counting and tracking farm animals using drone cameras. Reference [27] develops an algorithm for a drone to herd a flock of birds away from an airport. Field experiments show the effectiveness of such an algorithm. With the ability of rapidly crossing miles of rugged terrain, drones are potentially the ideal platforms for robotic animal herding, if they can efficiently interact with animals like sheepdogs. Sheepdogs usually herd animals by barking, glaring, or nibbling the heels of animals. For example, the New Zealand Huntaway uses its loud, deep bark to muster flocks of sheep [28]. Drones can act like sheepdogs by playing a pre-recorded dog bark loudly through a speaker, referred to as the barking drone. Recently, some successful attempts have been made using human-piloted barking drones to herd flocks of farm animals [29]. Besides, studies show that comparing with sheepdogs, using drones to herd cattle and sheep is faster and causes little pressure on animals [30].

1.1. Objectives and Contributions

This paper's primary objective is to design a robotic herding system that can efficiently herd a large number of farm animals without human input. The system should be able to collect a flock of farm animals when they are too dispersed and drive them to a designated location once they are aggregated. The main contributions of this paper are as follows:

- We propose a novel idea of autonomous barking drones by improving the design of the human-piloted barking drones and further propose a novel robotic herding system based on it. Comparing with the existing approaches of ground herding robots that drive animals through collisions or bright colours, the idea of autonomous barking drones can be a solution to the problem of effective robot-to-animal interaction with significantly improved efficiency;

- We propose a collision-free motion control algorithm for a network of barking drones to herd a large flock of farm animals efficiently;
- We conduct simulations of herding a thousand of animals, while the existing approaches usually herd tens of animals or swarm robots. The proposed algorithm can also be applied to herd swarm robots;
- Based on the animal behaviour model verified by real animal experiments and the proven shepherding examples by human-piloted barking drone, the proposed system has the potential to be the world's first practical robotic herding solution for a large flock of farm animals;
- With their functions being limited on non-essential applications such as animal monitoring and data collection, current Internet of Things (IoT) platforms for precision farming have a low return on investment. Besides solving the rigid demand (i.e., herding) for farmers, the proposed system can also serve as the IoT platform to achieve the same functions. Thus, it has the potential to popularise the IoT implementations for precision farming.

Preliminary versions of some results of this paper were presented in the PhD thesis of the first author [31].

1.2. Organization

The remainder of the paper is organised as follows. In Section 2, we introduce the design of the drone herding system. Section 3 presents the system model and problem statement. Drones motion control for gathering and driving is proposed in Section 4 and Section 5, respectively. Simulation results are presented in Section 6. Finally, we give our conclusions in Section 7.

2. Design of the Drone Herding System

We now introduce the proposed drone herding system. It consists of a fleet of two types of drones. The duty of the first type of drones is to detect and track animals. For this purpose, each drone is equipped with cameras and fitted with some Artificial Intelligence (AI) algorithms that can detect and track animals from live video feeds with a sufficient accuracy. The first type of drones shares some similarity with the goat tracking drones [21]. But different from it, our system only requires the tracking information of the animals on the boundary of the flock. This definitely relaxes the workload of the drones, and many existing image processing techniques, such as edge detection, can be adopted in real-time.

A drone of the second type is attached with a speaker that plays sheepdogs' barking. The speaker should have a clear voice, abundant volume, relatively small size and low weight. There have been some drone digital voice broadcasting systems on the market. For example, the MP130 from Chengzhi [32] can broadcast the voice for 500 m with a weight of 550 g. Moreover, the speaker is designed to be mounted on a stabiliser attached to the drone, so that it can stably broadcast to a desired direction, no matter which direction the barking drone is moving towards. It is worth to mention that the speaker on current human-piloted barking drone is not mounted on the stabiliser, so we improved the design of the current barking drones.

The observer and the barking drones have the communication ability. The communication between them can be realised by 2.4 gigahertz radio waves, which is commonly used by different drone products. The communication is mainly unidirectional from the observer to the barking drones. Specifically, the observer monitors the locations of animals on the edge of the flock and sends them to the barking drones in the real-time. A typical application scenario of the proposed system is herding a large flock of animals with one observer and multiple barking drones. Figure 1 shows a schematic diagram of a basic unit of the drone herding system, with one operator and one barking drone.

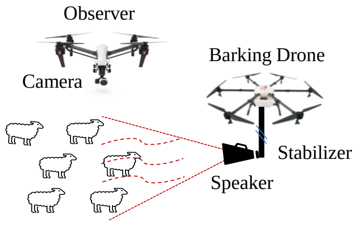


Figure 1. A basic unit of the proposed drone herding system.

Limited battery life is always a problem of drones applications. Later we will show that the proposed herding system can usually accomplish the herding task in less than 50 min, which is the common endurance of some commercialised industrial drone products such as the DJI M300 [33] (Note that, such industrial drones usually cost thousands of US dollars each, i.e., might be cheaper than a fully trained sheepdog). Moreover, any drone in the system should be able to autonomously fly back to a ground drone base station to recharge the battery with automatic charging devices [34]. Besides, the advancement of solar-harvesting technology enables the drones to prolong the battery lifetime [35].

3. System Model and Problem Statement

In this section, we introduce the dynamics models of animal motion and drone motion. Then, we present the herding problem formulation and the preliminaries for designing drones motion controller.

3.1. Animal Motion Dynamics

Same as most research on robotic herding [20,27,36], we describe the dynamic of animal flocking by boids model based on Reynolds' rules [37], which models the interaction of the agents based on attraction, repulsion and alignment models that are common in studies of collective animal behaviour. Field tests in [15] show that the boids model matches the behaviour of real flocks of sheep. The overall herd behaviour emerges from the decisions of individual members of the herd based on the four basic principles of collision avoidance, velocity matching (to nearby members of the herd), flock centring (the desire to be near the centre of the nearby members) and hazard avoidance. In detail, given the predefined constants r_s, N_s and r_a, an animal under consideration behaves according to the following rules:

1. If the distance to the barking drone is within r_s, then an animal will be repulsed directly away from the barking drone;
2. An animal is attracted to the centre of mass of its N_s nearest neighbours;
3. Animals are also locally repulsed from each other, so that if two or more animals are within a distance of r_a of each other there will be a repulsive force acting to separate them;
4. Otherwise, an animal is semistationary with some small random movements.

Let unit vectors $\vec{\aleph}_d$, $\vec{\aleph}_L$, $\vec{\aleph}_r$, $\vec{\aleph}_i$ and $\vec{\aleph}_e$ denote the repulsive force from the barking drone to the animal, the attractive force to the centre of mass of the animal's N_s nearest neighbours, the repulsive force from other animals within a distance of r_a of the animal, the inertial force to remain at the current location, and the noise element of the animal's movement, respectively. Then, the animal's moving direction vector $\vec{\aleph}$ is obtained by:

$$\vec{\aleph} = \eta_d \vec{\aleph}_d + \eta_L \vec{\aleph}_L + \eta_r \vec{\aleph}_r + \eta_i \vec{\aleph}_i + \eta_e \vec{\aleph}_e, \tag{1}$$

where η_d, η_L, η_r, η_i, and η_e are the weighting constants. Let V_{animal} be the maximum speed of the animal.

3.2. Drone Motion Dynamics

In this paper, we assume the barking drones maintain at a fixed altitude that is relatively low to keep the barking drones close to the herding animals. With the fixed altitude, we study the 2D motion of a barking drone described by the following mathematical model. Let

$$\boldsymbol{d}(t) := [x(t), y(t)] \tag{2}$$

be the 2D vector of the barking drone's Cartesian coordinates. Then, the motion of the barking drone is described by the equations:

$$\dot{\boldsymbol{d}}(t) = v(t)\boldsymbol{a}(t), \tag{3}$$

$$\dot{\boldsymbol{a}}(t) = \boldsymbol{u}(t), \tag{4}$$

where $\boldsymbol{a}(t) \in \mathbb{R}^2$, $|\boldsymbol{a}(t)| = 1$ for all t, $\boldsymbol{u}(t) \in \mathbb{R}^2$, $v(t) \in \mathbb{R}$, and the following constraints hold:

$$|\boldsymbol{u}(t)| \leq U_{max}, \quad v(t) \in [0, V_{max}], \tag{5}$$

$$(\boldsymbol{a}(t), \boldsymbol{u}(t)) = 0, \tag{6}$$

for all t. Here, $|\cdot|$ denotes the standard Euclidean vector norm, and $(\cdot,\cdot)$ denotes the scalar product of two vectors. The scalar variable $v(t)$ is the speed or linear velocity of the barking drone, and the scalar $\boldsymbol{u}(t)$ is applied to change the direction of the droneâ's motion, given by $\boldsymbol{a}(t)$. $v(t)$ and $\boldsymbol{u}(t)$ are two control inputs in this model. U_{max} and V_{max} are constants depending on the manufacturing of the drone. The condition (6) guarantees that the vectors $\boldsymbol{a}(t)$ and $\boldsymbol{u}(t)$ are always orthogonal. Furthermore, $\dot{\boldsymbol{d}}(t)$ is the velocity vector of the barking drone. The kinematics of many unmanned aerial vehicles can be described by the non-holonomic model (3)–(6); see, for example, [38] and references therein.

3.3. Problem Statement

Our goal is to navigate a network of barking drones to herd a flock of farm animals. A typical herding task consists of gathering and driving. In detail, we aim to navigate the barking drones to collect a flock of farm animals when they are too dispersed, namely gathering, and drive them to a designated location once they are aggregated, namely driving.

Let $\Phi = \{\boldsymbol{\phi}_i\}, i = 1, \ldots, n_s$ and $\mathcal{D} = \{\boldsymbol{d}_j\}, j = 1, \ldots, n_d$ be the sets of the two-dimensional (2D) positions of n_s farm animals and n_d barking drones, respectively. Let $C_o \in \mathbb{R}^2$ denote the position of the herding animals' centroid. For gathering, the goal is to gather all the herding animals around until the distance between C_o and any animal reaches a predefined constant R_c, that is,:

$$|\boldsymbol{\phi}_i C_o| \leqslant R_c, \forall i = 1, \ldots, n_s. \tag{7}$$

After gathering, the animals need to be driven to a designated location such as the centre of a sheepfold. The barking drones have to keep the animals aggregated during driving. Let G be the designated location. Similar to gathering, the goal of driving is formulated as:

$$|\boldsymbol{\phi}_i G| \leqslant R_c, \forall i = 1, \ldots, n_s. \tag{8}$$

3.4. Preliminaries

We now introduce the preliminaries for presenting the drones motion control algorithms, including the system's available measurement and the drones' motion restriction. During gathering, we use the convex hull of all the herding animals to describe the animals' footprint.

Available measurement: We assume that at any time t, the observer has the measurements of the positions of the vertices of the convex hull of all the herding animals, described by $\mathcal{P} = \{P_i\}, i = 1, \ldots, n_p$, and n_p is the number of vertices of the convex hull, which is much smaller than n_s. Besides, we assume the observer can estimate C_o with image processing techniques. The accurate real-time locations of the barking drones should also be available. In practice, the real-time drone locations can be provided by embedded GPS chips since the pastures are often open-air.

Definition 1. *The extended hull is a unique polygon surrounds the convex hull. The edges of the extended hull and the convex hull are in one-to-one correspondence, with each pair of the corresponding edges parallel to each other and maintaining the same fixed distance.*

Let $\mathcal{E} = \{E_i\}, i = 1, \ldots, n_e$ be the set of 2D positions of all the extended vertices in a counterclockwise manner, $n_e = n_p$. We now present the construction of the extended hull for a given convex hull of the herding animals. Let P_i be the position of a convex hull vertex with two neighbour vertices P_{i-1} and P_{i+1}. Construct parallel lines l_1 and l_2 of edges P_iP_{i+1} and P_iP_{i-1} on the periphery of the convex hull, respectively. Both the distance between l_1 and P_iP_{i+1} and the distance between l_2 and P_iP_{i-1} are d_s, where d_s is the predefined drone-to-animal distance. Let E_i be the position of the intersection of l_1 and l_2. Then, E_i is the extended hull's vertex corresponding to P_i, also called as the extended vertex of P_i. Let L_1 be the position of the intersection of l_1 and the extension line of P_iP_{i-1}. Let L_2 be the position of the intersection of l_2 and the extension line of P_iP_{i+1}, as shown in Figure 2. Naturally, $P_iL_1E_iL_2$ is a parallelogram. Let α denote $\angle L_1P_iL_2$. Then, we have:

$$\cos\alpha = \frac{(\overrightarrow{P_iP_{i-1}}, \overrightarrow{P_iP_{i+1}})}{|\overrightarrow{P_iP_{i-1}}||\overrightarrow{P_iP_{i+1}}|}, \tag{9}$$

$$\overrightarrow{P_iL_1} = \frac{d_s}{|\cos\alpha|} \cdot \frac{\overrightarrow{P_{i-1}P_i}}{|\overrightarrow{P_{i-1}P_i}|}, \tag{10}$$

$$\overrightarrow{P_iL_2} = \frac{d_s}{|\cos\alpha|} \cdot \frac{\overrightarrow{P_{i+1}P_i}}{|\overrightarrow{P_{i+1}P_i}|}. \tag{11}$$

Thus, given P_{i-1}, P_i and P_{i+1}, E_i can be computed by:

$$E_i = P_i + \overrightarrow{P_iL_1} + \overrightarrow{P_iL_2}. \tag{12}$$

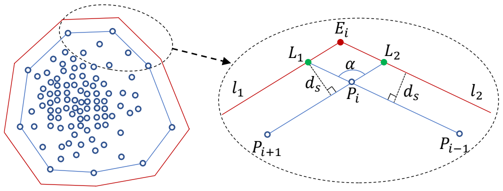
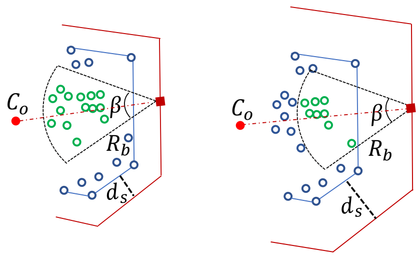


Figure 2. Illustration of the convex hull of the herding animals (○) and the corresponding extended hull.

Motion restriction: For the purpose of efficient gathering and avoiding to disperse any herding animals, all the barking drones are restricted to move only on the extended hull during gathering.

We assume that the spread range of the barking from the drone is fan-shaped, and only animals within this range will be affected by the repulsion of the barking drone. The intensity of barking outside this range is below the minimum level that can cause the evasive behaviour of the animal. We call this fan-shaped range as the barking cone, with the effective broadcasting angle β and distance R_b, respectively. The number of repulsed animals of a barking drone will be influenced by four aspects: β, R_b, drone-to-animal distance d_s and the distribution of animals. With the help of the stabiliser, the speaker should always face to C_o, as illustrated in Figure 3.



(**a**) (**b**)

Figure 3. Illustration of the effective broadcasting angle β and distance R_b with (**a**) a small drone-to-animal distance d_s and (**b**) a larger drone-to-animal distance d_s, under the same animals' distribution. Here (○) stands for the animal repulsing by the barking drone (■); (○) stands for the animal outsides the barking broadcasting range; (●) stands for C_o.

Remark 1. *Figure 3 also shows that a larger d_s may lead to fewer animals repulsing by the barking drone with the same β, R_b and animals' distribution, if fewer animals locating near the edges of the convex hull. But if there are more animals concentrated near the edges, the situation may be reversed.*

4. Drones Motion Control for Gathering

This section introduces the motion control algorithms for a network of barking drones to quickly accomplish the gathering task. We first introduce the algorithm for navigating the barking drones to fly to the extended hull and to fly on the extended hull in Section 4.1 and Section 4.2, respectively. Then, Section 4.3 presents the optimal positions (steering points) for the barking drones to efficiently gather animals, as well the collision-free allocation of the steering points. A flowchart of the proposed method is shown in Figure 4.

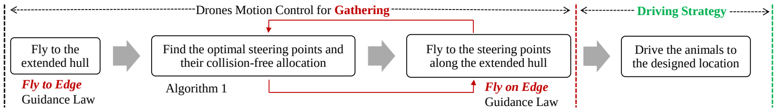

Drones Motion Control for Gathering
Driving Strategy
Fly to the extended hull
Fly to Edge
Guidance Law
Find the optimal steering points and their collision-free allocation
Algorithm 1
Fly to the steering points along the extended hull
Fly on Edge
Guidance Law
Drive the animals to the designed location

Figure 4. Overview of the proposed method.

Let A be any point on the plane of the extended hull. Let B be a vertex of the extended hull. We now introduce two guidance laws for navigating a barking drone from A to B in the shortest time:

1. Fly to edge: navigate the barking drone from an initial position A to the extended hull in the shortest time. Note that the vertices of the extended hull can be moving. Let O denote the barking drone's reaching point on the extended hull; see Figure 5.
2. Fly on edge: navigate the barking drone from O to B in the shortest time following a given direction, e.g., clockwise or counterclockwise, while keeping the barking drone on the extended hull. Since the drone has non-holonomic motion dynamics, it is allowed to move along a short arc when traveling between two adjacent edges, see Figure 5.

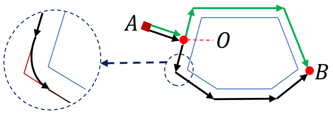


Figure 5. Illustration of the path planning for a barking drone from A to B, where A is a point on the plane of the extended hull, B is a vertex of the extended hull, O is the reaching point of the drone on the extended hull. The animals' convex hull is denoted by blue lines. The green arrows and black arrows are the planned trajectories with a given clockwise and counterclockwise direction, respectively.

Note that A is not necessarily outside the extended hull. To avoid dispersing any herding animal, the speaker on the barking drone should be turned on only when it has arrived at the extended hull. Besides, if A is already on the extended hull, let $O = A$ and apply *Fly on edge Guidance Law* directly.

4.1. Fly to Edge Guidance Law

Let $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ be non-zero two-dimensional vectors, and $|\boldsymbol{w}_1| = 1$. Now introduce the following function $F(\cdot,\cdot)$ mapping from $\mathbb{R}^2 \times \mathbb{R}^2$ to $\mathbb{R}^2$ as

$$\mathrm{F}(\boldsymbol{w}_1, \boldsymbol{w}_2) := \begin{cases} 0, & f(\boldsymbol{w}_1, \boldsymbol{w}_2) = 0 \\ |f(\boldsymbol{w}_1, \boldsymbol{w}_2)|^{-1} f(\boldsymbol{w}_1, \boldsymbol{w}_2), & f(\boldsymbol{w}_1, \boldsymbol{w}_2) \neq 0 \end{cases}, \tag{13}$$

where $f(\boldsymbol{w}_1, \boldsymbol{w}_2) := \boldsymbol{w}_2 - (\boldsymbol{w}_1, \boldsymbol{w}_2)\boldsymbol{w}_1$. In other words, the rule (14) defined in the plane of vectors $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$. The resulted vector $F(\boldsymbol{w}_1, \boldsymbol{w}_2)$ is orthogonal to $\boldsymbol{w}_1$ and directed "towards" $\boldsymbol{w}_2$. Moreover, introduce the function $g(\boldsymbol{w}_1, \boldsymbol{w}_2)$ as follows

$$g(\boldsymbol{w}_1, \boldsymbol{w}_2) := \begin{cases} 1, (\boldsymbol{w}_1, \boldsymbol{w}_2) > 0, \\ -1, (\boldsymbol{w}_1, \boldsymbol{w}_2) \leq 0, \end{cases} \tag{14}$$

We will also need the following notations to present the *Fly to edge* guidance law. At time t, let $E_{j+1}E_j$ be the extended hull edge that is the closest to the drone. We will show how to find $E_{j+1}E_j$ later. Let $\boldsymbol{q}(t) \in \mathbb{R}^2$ denote the vector from vertex E_{j+1} to E_j. Let $\boldsymbol{p}(t) \in \mathbb{R}^2$ denote the vector from the vertex E_{j+1} to the drone. Let O be the point on $E_{j+1}E_j$ that is the closest to the drone. Let $\boldsymbol{b}(t)$ be the vector from the drone to O. If $(\boldsymbol{p}, \boldsymbol{q}) < 0$, we have $O = E_{j+1}$ and $\boldsymbol{b}(t) = -\boldsymbol{p}(t)$, see Figure 6a. If $(\boldsymbol{p}, \boldsymbol{q}) > 0$ and $|\boldsymbol{q}| \geqslant |\boldsymbol{p}|$, $\boldsymbol{b}(t)$ is always orthogonal to $\boldsymbol{q}(t)$. Let $\boldsymbol{o}(t)$ be the vector from E_{j+1} to O, see Figure 6b. $\boldsymbol{o}(t)$ can be obtained by the following equation:

$$|\boldsymbol{o}(t)| = |\boldsymbol{q}(t)|^{-1}(\boldsymbol{p}(t), \boldsymbol{q}(t)), \tag{15}$$

$$\boldsymbol{o}(t) = |\boldsymbol{o}(t)| \cdot |\boldsymbol{q}(t)|^{-1}\boldsymbol{q}(t), \tag{16}$$

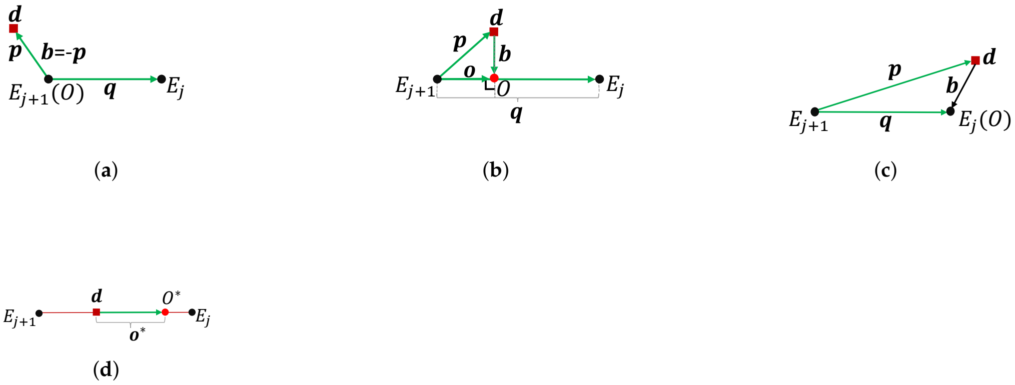


Figure 6. Illustration of (**a**) *Fly to edge* guidance with $(\boldsymbol{p}, \boldsymbol{q}) < 0$; (**b**) *Fly to edge* guidance with $(\boldsymbol{p}, \boldsymbol{q}) < 0$ and $|\boldsymbol{q}| \geqslant |\boldsymbol{p}|$; (**c**) *Fly to edge* guidance otherwise; (**d**) *Fly on edge* guidance navigates the drone from $\boldsymbol{d}$ to O^*.

Otherwise, we have $O = E_j$ and $\boldsymbol{b}(t) = \boldsymbol{q}(t) - \boldsymbol{p}(t)$, see Figure 6c. Given $\boldsymbol{p}(t)$ and $\boldsymbol{q}(t)$, we present the following *Fly to edge* guidance law:

$$\boldsymbol{b}(t) = \begin{cases} -\boldsymbol{p}(t), & \text{if } (\boldsymbol{p}, \boldsymbol{q}) < 0, \\ \boldsymbol{o}(t) - \boldsymbol{p}(t), & \text{if } (\boldsymbol{p}, \boldsymbol{q}) > 0 \text{ and } |\boldsymbol{p}| > |\boldsymbol{q}|, \\ \boldsymbol{q}(t) - \boldsymbol{p}(t), & \text{otherwise}, \end{cases} \tag{17}$$

$$\boldsymbol{u}(t) = U_{max} g(\boldsymbol{a}(t), \boldsymbol{b}(t)) F(\boldsymbol{a}(t), \boldsymbol{b}(t)), \tag{18}$$

$$v(t) = V_{max} g(\boldsymbol{a}(t), \boldsymbol{b}(t)). \tag{19}$$

The proposed *Fly to edge* guidance law belongs to the class of sliding-mode control laws (see, e.g., [39]). With the simple switching strategy, sliding mode control laws are quite robust and not sensitive to parameter variations and uncertainties of the control channel. Moreover, because the control input is not a continuous function, the sliding mode can be reached in finite time which is better than asymptotic behaviour, see for example, [39–41].

Assumption 1. *Let $\mathcal{X}(t)$ be the length of $E_iE_(i-1)$ at time t. Then $\mathcal{X}_1 \leq \mathcal{X}(t) \leq \mathcal{X}_2$ for all t for some given constants $0 < \mathcal{X}_1 < \mathcal{X}_2$. Let D_1 and D_2 be some constants such that $0 < D_1 < D_2 < \mathcal{X}_1$. Let $\delta(t)$ be the distance between the drone and E_j. $\delta(0)$ is the distance between A and E_j.*

Assumption 2. *Let $\xi := \min\left\{\frac{\mathcal{X}_1 - D_2}{D_2}, \frac{D_1}{\mathcal{X}_2 - D_1}\right\}$. Then*

$$\xi V_{\max} > \frac{D_1 V_{animal}}{2\mathcal{X}_1}, \tag{20}$$

$$U_{\max} > \frac{V_{animal}}{\mathcal{X}_1}, \quad V_{\max} > \frac{(\mathcal{X}_2 - D_1) V_{animal}}{\mathcal{X}_1}, \tag{21}$$

$$D_1 + \frac{\pi(V_{\max} + V_{animal})}{U_{\max}} \leq \delta(0) \leq D_2 - \frac{\pi(V_{\max} + V_{animal})}{U_{\max}}. \tag{22}$$

Theorem 1. *Suppose that Assumptions 1 and 2 holds. Then, the guidance law (17), (18) and (19) navigate the barking drone from an initial position A to $E_{j+1}E_j$ and remains on $E_{j+1}E_j$ after arrival.*

Remark 2. *It should be pointed out that Assumptions 1 and 2 are quite technical assumptions, which are necessary for a mathematically rigorous proof of the performance of the proposed guidance law. However, our simulations show that the proposed guidance law often performs well even in situations when Assumptions 1 and 2 do not hold.*

Proof. From the definitions of $F(\boldsymbol{w}_1, \boldsymbol{w}_2)$ and $g(\boldsymbol{w}_1, \boldsymbol{w}_2)$, the guidance law (18), (19) turns the velocity vector $\dot{\boldsymbol{d}}(t)$ towards $\boldsymbol{b}(t)$. Moreover, Equation (17) gives that $\boldsymbol{b}(t)$ is pointing from the drone to its closest point on $E_{j+1}E_j$, see Figure 6a–c. Furthermore, it follows from (22) together with Assumption 1 that there exists some time $t_\star$ that for all $t \geq t_\star$, the vectors $\dot{\boldsymbol{d}}(t)$ and $\boldsymbol{b}(t)$ are co-linear and $D_1 \leq \delta(t) \leq D_2$. Introduce the function $Y(t) = |\boldsymbol{b}(t)|$ is the distance between the drone's current location and $E_{j+1}E_j$. Then, it follows from (20) of Assumption 2 and the inequality $D_1 \leq \delta(t) \leq D_2$ that if $Y(t) > 0$ then $\dot{Y}(t) < -\tau$ for some constant $\tau > 0$. Therefore, there exists a time t_a such that the drone's current position belongs to $E_{j+1}E_j$ for all $t \geq t_a$. Moreover, (21) implies that for all $t \geq t_a$ the drone will remain in the sliding mode of the system (3), (4), (17), (18), (19) corresponding to the position of the drone on $E_{j+1}E_j$, and the and the vector $\dot{\boldsymbol{d}}(t)$ orthogonal to $E_{j+1}E_j$. This completes the proof of Theorem 1. □

Remark 3. *At time t, given $\boldsymbol{d}(t)$ and E, calculate $\boldsymbol{b}(t)$ for the barking drone to each edge of the extended hull. Then, the edge with the minimum $|\boldsymbol{b}(t)|$ is the closest edge of the extended hull to the drone.*

4.2. Fly on Edge Guidance Law

Before introducing the *Fly on edge* guidance law, we first present the *edge sliding* guidance law for a drone flying along an edge of the extended hull, with possibly moving vertices. At time t, let $E_{j+1}E_j$ be the edge that we want to keep the drone on. Let $O^* \in E_jE_{j+1}$ be the target position of the barking drone. Let $\boldsymbol{o}^*(t) \in \mathbb{R}^2$ denote the vector from the drone to O^*, as shown in Figure 6d. We introduce $\boldsymbol{b}^*(t)$ that is given by:

$$\boldsymbol{b}^*(t) = \boldsymbol{b}(t) + \boldsymbol{o}^*(t). \tag{23}$$

Then, the *edge sliding* guidance law is as follows:

$$\boldsymbol{u}(t) = U_{max} g(\boldsymbol{a}(t), \boldsymbol{b}^*(t)) F(\boldsymbol{a}(t), \boldsymbol{b}^*(t)), \tag{24}$$

$$v(t) = V_{max} g(\boldsymbol{a}(t), \boldsymbol{b}^*(t)). \tag{25}$$

Theorem 2. *Suppose that Assumption 1 holds. Then the guidance law (23), (24) and (25) navigates the barking drone from $\boldsymbol{d}(t)$ to O^* along $E_{j+1}E_j$, and enables the drone to stay at O^*.*

Note that, $\boldsymbol{b}^*(t)$ consists of two vector components: $\boldsymbol{b}(t)$ and $\boldsymbol{o}^*(t)$. Where $\boldsymbol{b}(t)$ is for keeping the drone on $E_{j+1}E_j$ and $\boldsymbol{o}^*(t)$ is for navigating the drone to O^*. Thus, Theorem 2 can be proved similar to Theorem 1.

Suppose that the barking drone has arrived at the point O at time $t = t_0$. Introduce a direction index $\sigma = 0$ for clockwise flying and $\sigma = 1$ for counterclockwise flying. Given O, $\mathcal{E}$, B and σ, the *Fly on edge* guidance law solves for the barking drone's control input $v(t)$ and $\boldsymbol{u}(t)$ by commanding one of the two following motions.

4.2.1. TRANSFER

The drone flies to the adjacent edge in the given direction from its current location through a straight line and an arc. Consider that the drone flies from its current edge $E_{j-1}E_j$ to the adjacent edge E_jE_{j+1}. The drone first moves to $F_1 \in E_{j-1}E_j$ following the *edge sliding* guidance law. Then, let $\boldsymbol{u}(t) = U_{max}F(\boldsymbol{a}(t), \boldsymbol{e}(t))$ and $v(t) = V_{max}$, where $\boldsymbol{e}(t) := \overrightarrow{\boldsymbol{d}(t)C_t}$ denote the vector points from the drone's current location to the turning centre C_t. The drone will turn with the minimum turning radius and arrives at $F_2 \in E_jE_{j+1}$, as shown in Figure 7.

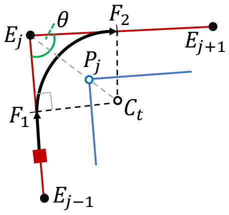


Figure 7. Illustration of the motion *TRANSFER*.

Let ρ be the minimum turning radius of the drone. Let $\theta \in (0, \pi)$ denote the angle $\angle E_{j-1}E_jE_{j+1}$. P_j is the convex hull vertex corresponding to E_j. Then, F_1 and C_t can be computed by:

$$F_1 = E_j + \frac{\overrightarrow{E_jE_{j-1}}}{|\overrightarrow{E_jE_{j-1}}|} \cdot \frac{\rho}{\tan\frac{\theta}{2}}, \tag{26}$$

$$C_t = P_j + \frac{\overrightarrow{E_jP_j}}{|\overrightarrow{E_jP_j}|} \cdot \frac{\rho - d_s}{\sin\frac{\theta}{2}}, \tag{27}$$

where d_s denotes drone-to-animal distance, as defined above. θ can be obtained from the equation similar to (10). To avoid dispersing any animals, the turning trajectory should not touch the convex hull of the animals, which always holds in the case of $d_s > \rho$. If $d_s \leq \rho$, the following inequalities need to be satisfied:

$$\begin{aligned} &|C_tP_j| = \tfrac{\rho - d_s}{\sin\frac{\theta}{2}} < \rho, \\ \Rightarrow\ &\sin\tfrac{\theta}{2} > 1 - \tfrac{d_s}{\rho}, \\ \Rightarrow\ &\theta > 2\arcsin\left(1 - \tfrac{d_s}{\rho}\right). \end{aligned} \tag{28}$$

Remark 4. *If (28) does not hold, the drone directly flies to E_j following the* edge sliding *guidance law, then stops and changes direction to $\overrightarrow{E_jE_{j+1}}$, which is slower than along the arc trajectory. An isolated animal far away from all the other animals may cause a very small θ and lead to this problem.*

4.2.2. BRAKE

If the drone has arrived at an edge E_bB, where $E_b \in \mathcal{E}$ is the closest vertex of B on the opposite of the direction indicated by σ, then the drone flies to a point B following the *edge sliding* guidance law. Let $E'_0 \in \mathcal{E}$ be the closest vertex to the drone opposite to the given direction. Let $\mathcal{H} \subseteq \mathcal{E}$ be the set of vertices locating between O and B along the given direction.

We are now in a position to present the *Fly on edge* guidance law, as shown in Algorithm 1. Specifically, the drone approaches the edge E_bB that contains the destination B by performing *TRANSFER* multiple times. Afterwards, the drone starts *BRAKE* once $\mathcal{H}$ is found to be an empty set (i.e., $\mathcal{H} = \emptyset$), which means the drone has arrived at E_bB. The drone will then reaches B through *BRAKE*. The presented guidance law is designed to navigate the barking drone from any point on the extended hull to a selected vertex following a given direction, and stop the drone at the selected vertex.

Algorithm 1: Fly On Edge Guidance Law.

```
   Input $O, \mathcal{E}, B, \sigma, \rho$
1: Find $E'_0$ and $\mathcal{H}$, if $\mathcal{H} = \emptyset$, go to line 4;
2: $[v(t), \boldsymbol{u}(t)]$ = $TRANSFER(E'_0, \mathcal{H})$;
3: Repeat lines 1, 2;
4: $[v(t), \boldsymbol{u}(t)]$ = $BRAKE(E'_0, B)$.
```

4.3. Selection and Allocation of Steering Points

We now find the optimal positions for the barking drones to effectively gather animals. Aiming to minimise the maximum animal-to-centroid distance in the shortest time, at any time t, we choose the animals with the largest animal-to-centroid distance as the target animals. These animals are also the convex hull vertices that are farthest to C_o. Since the barking drones have their motions restricted on the extended hull, we select the extended hull vertices corresponding to the target animals as the optimal drone positions for steering the target animals to approach C_o. From now on, we call these corresponding extended hull vertices the steering points, denoted by the set $\mathcal{S} = \{\boldsymbol{s}_j\}, j = 1, \ldots, n_d, \mathcal{S} \subseteq \mathcal{E}$.

Remark 5. *For a large flock of animals, $n_e \geq n_d$ holds generally. In the cases of $n_e < n_d$, let $n_d - n_e$ barking drones that are far from the steering points quit the gathering task, stand by at their current locations. These drones may rejoin the gathering task when n_e increases afterwards.*

Definition 2. *The allocation of steering points specifies which drone goes to which steering point through which direction, that is, clockwise or counterclockwise.*

The optimal allocation of steering points should meet the following two requirements:

1. No collision happens when each drone is flying to its allocated steering point along the extended hull;
2. With requirement 1 met, the maximum travel distance of the drones is minimised.

Suppose that all the drones have arrived at the extended hull at time $t = t_1$. We relabel the drones so that the index of the drone increases in the counterclockwise direction. Let M be the perimeter of the extended hull. Imagine that we disconnect the extended hull from the position of the first drone, that is, $\boldsymbol{d}_1(t)$. Then, 'straighten' the extended hull into a straight line segment with a length of M, so that $\mathcal{D}$, $\mathcal{S}$ and $\mathcal{E}$ become the points on the line segment. Based on this line segment we build a one-dimensional (1D) coordinate axis denoted as the z axis. Let $\mathcal{Z} = \{z_j\}, j = 1, \ldots, n_d$ be the 1D coordinates of the drones' positions on the z axis. Let $z_1 = 0$ be the origin. We have $z_j < z_{j+1}, j = 1, \ldots, n_d - 1$, as shown in Figure 8a. It can be seen that the left and right flying on the z axis corresponding to counterclockwise and clockwise flying on the extended hull, respectively.

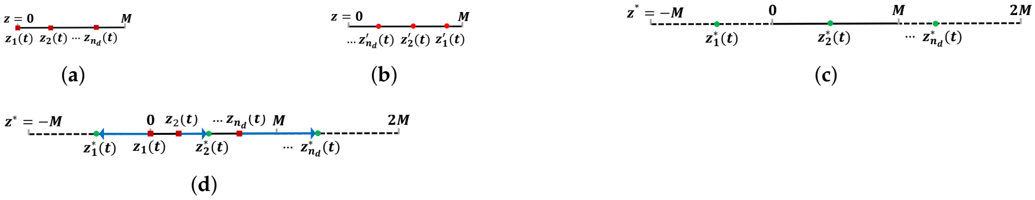


Figure 8. The examples of (**a**) drones' positions on the z axis, (**b**) steering points' positions on the z axis, (**c**) steering points on the z^* axis, and (**d**) drone's positions, steering points' positions and travel routes on the z^* axis. Here (■) stands for z_j, (●) stands for z'_j, (●) stands for z^*_j, and the blue arrows stand for the drones' travel routes.

We will also need the following notations to present our algorithm. Let $\mathcal{S}' = \left\{\boldsymbol{s}'_j\right\}$, $j = 1, \ldots, n_d$ be a set of *allocated* steering points with a corresponding z axis coordinates set $\mathcal{Z}' = \left\{z'_j\right\}, j = 1, \ldots, n_d$, as shown in Figure 8b. $\mathcal{Z}'$ is the destination of the drones on the z axis. Note that, $z'_j < z'_{j+1}, j = 1, \ldots, n_d - 1$ may not hold. Let $\Gamma = \{\gamma_j\}, j = 1, \ldots, n_d$ be the set of the drones' travel distances for reaching their allocated steering points. We now define three variables σ_j, λ^R_j and $\lambda^L_j \in \{0, 1\}$ to indicate the flying direction and extent of drone j. Specifically, let $\sigma_j = 1$ if drone j reaches $z'_j(t)$ by right flying on the z axis, and $\sigma_j = 0$ if drone j reaches $z'_j(t)$ by left flying on the z axis. Furthermore, let $\lambda^R_j = 1$ if drone j will pass $z = 0$ by right flying to reach $z'_j(t)$, and $\lambda^R_j = 0$ otherwise. Similarly, let

$\lambda_j^L = 1$ if drone j will pass $z = 0$ by left flying to reach $z'_j(t)$, and $\lambda_j^L = 0$ otherwise. Let $\Sigma = \{\sigma_j\}$, $\Lambda^L = \left\{\lambda_j^L\right\}$ and $\Lambda^R = \left\{\lambda_j^R\right\}$, $j = 1, \ldots, n_d$ be the sets of λ_j^L and λ_j^R, respectively. Given z_j, z'_j and σ_j, λ_j^L and λ_j^R can be computed by:

$$\lambda_j^L = \begin{cases} 1, & \text{if } z'_j > z_j \text{ and } \sigma_j = 0, \\ 0, & \text{otherwise,} \end{cases} \tag{29}$$

$$\lambda_j^R = \begin{cases} 1, & \text{if } z'_j < z_j \text{ and } \sigma_j = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{30}$$

The main notations are listed in Table 1. Since the line segment $z = [0, M]$ is generated by straightening the enclosed extended hull, the drones passed $z = 0$ by left flying will appear on the right side of the line segment, and the drones passed $z = M$ by right flying will be appear on the left side of the line segment. We now imagine extending the line segment $z = [0, M]$ to $z = [-M, 2M]$ and build another 1D coordinate z^* axis as shown in Figure 8c. On the z^* axis, the drones passed $z^* = 0$ by left flying will not appear on the right side of the line segment, but will appears on $z^* = [-M, 0]$, and the drones passed $z^* = M$ by right flying will be appear on $z^* = [M, 2M]$. Let $\mathcal{Z}^* = \left\{z_j^*\right\}$, $j = 1, \ldots, n_d$ be the 1D coordinates set on the z^* axis corresponding to $\mathcal{Z}'$. Then, the mapping between $\mathcal{Z}^*$ and $\mathcal{Z}'$ is obtained by:

$$z_j^* = \begin{cases} z'_j - \lambda_j^L M, & \text{if } \sigma_j = 0, \\ z'_j + \lambda_j^R M, & \text{if } \sigma_j = 1. \end{cases} \tag{31}$$

If place $\mathcal{Z}$ on the z^* axis, as shown in Figure 8c, the travel route of any drone j will be $\overrightarrow{z_j z_j^*}$. We obtain the expression for the travel distances γ_j as follows:

$$\gamma_j = \begin{cases} z_j - z_j^*, & \text{if } \sigma_j = 0, \\ z_j^* - z_j, & \text{if } \sigma_j = 1. \end{cases} \tag{32}$$

Then, the steering points allocation optimization problem is formulated as follows:

$$\min_{S', \Sigma} \max_{j=1,\ldots,n_d} \gamma_j, \tag{33}$$

s.t.

$$0 < z_{n_d}^* - z_1^* < M, \tag{34}$$

$$z_j^* < z_{j+1}^*, \; j = 1, \ldots, n_d - 1, \tag{35}$$

where (33) minimises the travel distance of the drone farthest to its allocated steering point.

Table 1. Notations and Descriptions.

Notation	Description
M	Perimeter of the extended hull.
$\mathcal{D} = \left\{ \boldsymbol{d}_j \right\}$	Set of the drones' 2D positions.
$\mathcal{Z} = \left\{ z_j \right\}$	Set of the z coordinates of $\mathcal{D}$.
$\mathcal{S} = \left\{ \boldsymbol{s}_j \right\}$	Set of the steering points' 2D positions.
$\mathcal{S}' = \left\{ \boldsymbol{s}'_j \right\}$	Set of the allocated steering points' 2D positions.
$\mathcal{Z}' = \left\{ z'_j \right\}$	Set of the z coordinates of $\mathcal{S}'$.
$\mathcal{Z}^* = \left\{ z^*_j \right\}$	Set of the z^* coordinates of $\mathcal{Z}'$.
$\Gamma = \left\{ \gamma_j \right\}$	Set of drones' travel distances .
$\Sigma = \left\{ \sigma_j \right\}$	Set of drones' flying directions.
$\Lambda^L = \left\{ \lambda^L_j \right\}$	Set of the indicators of passing $z = 0$ by right flying.
$\Lambda^R = \left\{ \lambda^R_j \right\}$	Set of the indicators of passing $z = 0$ by left flying.

Assumption 3. *All the drones start flying to their allocated steering points at the same time, follow the proposed* Fly on edge *guidance law.*

Theorem 3. *Suppose that Assumption 2 holds. Then, (34), (35) guarantees that no collision happens when the drones are flying to their allocated steering points.*

Proof. Suppose that all the drones start flying to their steering points at time $t = t_0$. Let $t = t^j_f$ be the time of drone j arrives at its steering point, that is, $z^*_j(t)$. From (14) and (25), at any time $t = t_s \in [t_0, t^j_f]$, drone $j \in \{2, \ldots, n_d - 1\}$ has:

$$|v(t_s)| = V_{max}. \tag{36}$$

From the proof of Theorem 1, $\boldsymbol{b}(t)$ is always been minimised after drone j arrived at the extended hull. Since drone j is moving from $z_j(t_0)$ to z^*_j along the z axis, it can be obtained from (23) and (3) that

$$\boldsymbol{b}^*(t_s) = z^*_j - z_j(t_s), \tag{37}$$

$$z_j(t_s) = z_j(t_0) + (t_s - t_0)V_{max}, \tag{38}$$

$$z^*_j = z_j(t_0) + (t^j_f - t_0)V_{max}. \tag{39}$$

Then, the distance between drone j and drone $j+1$ at time $t = t_s \in [t_0, \max(t^j_f, t^{j+1}_f)]$ can be computed by

$$z_{j+1}(t_s) - z_j(t_s) = \begin{cases} z_{j+1}(t_0) - z_j(t_0), & \text{if } t_s \le \min(t^j_f, t^{j+1}_f), \\ z_{j+1}(t_s) - z^*_j, & \text{if } t^j_f \le t^{j+1}_f \text{ and } t^j_f < t_s \le t^{j+1}_f, \\ z^*_{j+1} - z_j(t_s), & \text{if } t^j_f > t^{j+1}_f \text{ and } t^{j+1}_f < t_s \le t^j_f. \end{cases} \tag{40}$$

Since $z_j < z_{j+1}, j = 1, \ldots, n_d - 1$, it can be concluded from (35), (38), (39) and (40) that

$$z_{j+1}(t_s) - z_j(t_s) > 0, \quad t_s \in [t_0, \max(t_f^j, t_f^{j+1})] \tag{41}$$

Which means drone $j \in \{2, \ldots, n_d - 1\}$ will not collide with drone $j + 1$ before they arrived at their steering points. Moreover, the actual distance $|z_1 z_{n_d}|$ between drone 1 and drone n_d is given by

$$|z_1 z_{n_d}| = \begin{cases} z_{n_d}(t_s) - z_1(t_s), \text{ if } z_{n_d}(t_s) - z_1(t_s) \leq M/2, \\ M - (z_{n_d}(t_s) - z_1(t_s)), \text{ if } z_{n_d}(t_s) - z_1(t_s) > M/2. \end{cases} \tag{42}$$

Given (34), $|z_1 z_{n_d}| > 0, t_s \in [t_0, \max(t_f^1, t_f^{n_d})]$ can be proved similarly. Therefore, (34) guarantees that drone 1 will not collide with drone n_d, and (35) guarantees that each drone will not collide with their neighbours. This completes the proof of Theorem 3. □

Remark 6. *For n_d drones, n_d steering points and two possible directions for each drone, the number of possible allocations is $N = n_d!2^{n_d}$.*

Since n_d is often a limited number, N will be limited as well. Therefore, the optimal allocation can be found by generating and searching all the possible allocations. We are now in a position to present the algorithm to find the optimal steering points allocation, as shown in Algorithm 2.

Algorithm 2: Optimal Steering Points Allocation.

```
Input $n_d$, $E$, $D$
Find $\mathcal{S} \in \mathcal{E}$;
Calculate $\mathcal{Z}$ from $\mathcal{D}$ and $\mathcal{E}$;
Generate possible allocations $(\mathcal{S}', \Sigma)^k,\ k = 1, \ldots, N$;
For each $(\mathcal{S}', \Sigma)^k$, calculate $(\mathcal{Z}^*, \Gamma)^k$;
Solve (29)–(31) by searching $(\mathcal{Z}^*, \Gamma)^k,\ k = 1, \ldots, N$.
```

Suppose that the gathering task starts at $t = 0$. The proposed herding system first navigates all the barking drones to the extended hull by *Fly to edge* guidance law. Then, the system calculates the optimal steering points allocation after every sampling time Δ and navigates the barking drones to their allocated steering points by *Fly on edge* guidance law, until (7) is satisfied, that is, $|\boldsymbol{\phi}_i C_o| \leqslant R_c, \forall i = 1, \ldots, n_s$. It is worth mentioning that the optimal allocation may change before some drones arrived at their allocated steering points due to the animals' movement. The gathering task, however, will be interrupted. Because as long as the barking drones are flying on the extended hull, the animals inside the drones' barking cone will be repulsed to move towards C_o.

5. Drones Motion Control for Driving

Suppose that (7) is satisfied at $t = t_g$, the goal is then transferred to drive the gathered animals to a desired location, for example, the centre of a sheepfold. The convex hull of the gathered animals will be close to a circle. For simplicity, from now on, we use the smallest enclosing circle to describe the footprint of the gathered animals. Let R_c' and C_o' be the radius and centre of the animals' smallest enclosing circle during driving, respectively. Similar to the definition of the extended hull, we define the *extended circle* as a circle with a larger radius $R_e := R_c' + d_s$ centred at C_o'.

According to (7), $R_c' = R_c$ and $C_o' = C_o$ when $t = t_g$. Imagine a point $C_* \in C_o G$ is moving from C_o to G with a constant speed $V_{driving} \leq V_{animal}$ when $t_g \leq t \leq t_d$, where t_d

denotes the time when the driving task is finished (i.e., (8) is satisfied). Given $C_o = [x_o, y_o]$ and $G = [x_g, y_g]$, $C_* = [x_*, y_*]$ can be computed by:

$$\begin{bmatrix} x_* \\ y_* \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + V_{driving}\left(t - t_g\right)\frac{\overrightarrow{CG}}{|\overrightarrow{CG}|}, t_g \le t \le t_d. \tag{43}$$

We aim to drive the animals so that C_o' can follow C_* moving from C_o to G, with $V_{driving}$ as the driving speed. Note that, a smaller $V_{driving}$ is preferred for a larger animals' number n_s, because a larger flock of animals tend to move slower. To this end, we adopt a side-to-side movement for the baking drones, which is a common animal driving strategy that can also be seen in [15,42], and so forth. Let l_g be the perpendicular line of C_oG that passes C_*. Let $\mathcal{L}$ be the semicircle of the extended circle cut by l_g that is farther to G; see Figure 9. Let $\mathcal{Q} = \{Q_j\}, j = 1, \ldots, n_d + 1$ be the set of points that *evenly* distributed on $\mathcal{L}$. Each drone j is then deployed to fly on $\mathcal{L}$ with Q_j and Q_{j+1} as its start and end points, respectively, as shown in Figure 9. With C_* approaching G, the side-to-side movements of the barking drones can 'push' the animals to approach G while keeping them aggregated.

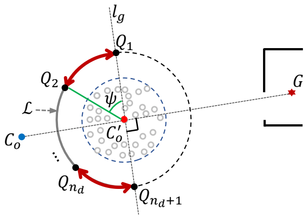


Figure 9. Barking drones deployment for animal driving, where the star marker stands for the designated location G; the red dot stands for C_o'. The dark red arrows stand for the side-to-side trajectories of the barking drones.

Given R_e and $C_* = [x_*, y_*]$, $Q_j := [x_j^q, y_j^q], j = 1, \ldots, n_d + 1$ can be computed by solving the following equations:

$$\left(x_j^q - x_*\right)^2 + \left(y_j^q - y_*\right)^2 = R_e^2, \tag{44}$$

$$k_1\left(x_{1,n_d+1}^q - x_*\right) + k_2\left(y_{1,n_d+1}^q - y_*\right) = 0. \tag{45}$$

If n_d is an odd number,

$$\begin{aligned} &k_1\left(x_{j,n_d-j+2}^q - x_*\right) + k_2\left(y_{j,n_d-j+2}^q - y_*\right) \\ &= \mu R_e \cos\left(\frac{\pi}{2} + \psi(j-1)\right), \quad j = 2, \ldots, \frac{n_d+1}{2}. \end{aligned} \tag{46}$$

If n_d is an even number,

$$\begin{aligned} &k_1\left(x_{j,n_d-j+2}^q - x_*\right) + k_2\left(y_{j,n_d-j+2}^q - y_*\right) \\ &= \mu R_e \cos\left(\frac{\pi}{2} + \psi(j-1)\right), \quad j = 2, \ldots, \frac{n_d}{2}, \end{aligned} \tag{47}$$

$$k_1\left(x_{n_m}^q - x_*\right) + k_2\left(y_{n_m}^q - y_*\right) = -\mu R_e, \tag{48}$$

where

$$\psi = \frac{\pi}{n_d}, \tag{49}$$

$$k_1 = x_g - x_*, \quad k_2 = y_g - y_*, \tag{50}$$

$$\mu = \sqrt{{k_1}^2 + {k_2}^2}. \tag{51}$$

$$n_m = \frac{n_d}{2} + 1. \tag{52}$$

Specifically, once (7) is satisfied, all the barking drones immediately fly to the extended circle following the guardians law similar to the *Fly to edge* introduced in Section 4.1. It is worth mentioning that the extended polygon is inscribed in the extended circle, so the process of the drones flying to their closest point on the extended circle will not disperse any animal. After reaching the extended circle, the barking drones fly to their allocated start points in $\mathcal{Q}^* = \{Q_j\}, j = 1, \ldots, n_d \in \mathcal{Q}$ following the guardians law similar to the *Fly on edge* introduced in Section IV.2. The allocation of $\mathcal{Q}^*$ can be found by the algorithm similar to Algorithm 2 introduced in Section IV.3. Then, drone j continuously flies between Q_j and Q_{j+1} along $\mathcal{L}$, as shown in Figure 9, until (8) is satisfied.

6. Results

In this section, the performance of the proposed method is evaluated using MATLAB. Each simulation runs for 20 times. The animal motions dynamic parameters are chosen based on the field tests with real sheep conducted by [15], as shown in Table 2. Table 2 also shows some parameters of the barking drones, if not specified in the following part.

Table 2. Simulation Parameter Values.

Parameters	Values	Parameters	Values
N_s	20	r_a	2 m
η_d	1	η_L	1.05
η_r	2	η_i	0.5
η_e	0.3	Δ	0.2 s
V_{max}	25 m/s	U_{max}	5 m/s^2
V_{animal}	4 m/s	β	$\frac{2\pi}{3}$
R_b	100 m	d_s	30 m
n_s	200, 1000	R_c	60, 110 m
$V_{driving}$	3.8 m/s, 1.9 m/s	n_d	4

For comparison, we introduce an intuitional collision-free method as the benchmark method. Specifically, the benchmark method divides the extended hull into n_d segments with the same length at any time during gathering. Each drone is allocated to a segment and do the aforementioned side-to-side movement on the extended hull, until (7) is satisfied. The benchmark method adopts the same driving strategy as the proposed method. We consider that the animals are randomly distributed in an area with the size of 1200 m by 600 m as the initial field.

We first present some illustrative results showing 4 barking drones on two cases herding 200 and 1000 animals, respectively; see https://youtu.be/KMWxrlkU6t0, (accessed on 8 december 2021) and https://youtu.be/KPGrAcgPH8Q, (accessed on 8 december 2021). We can observe that the proposed method completes the gathering task in 9.5 min for the instance with 200 animals and 10.1 min for the case with 1000 animals. The total time for gathering and driving are 15 and 18.2 min for these cases. However, the benchmark method uses about 4.9 and 4.1 more minutes to complete these missions. Figure 10a shows how the animals' footprint radius changes with time t for these cases. We also present snapshots when $t = 0$, $t = 5$ and $t = 8$ min for the case of 1000 animals in Figure 10b–f.

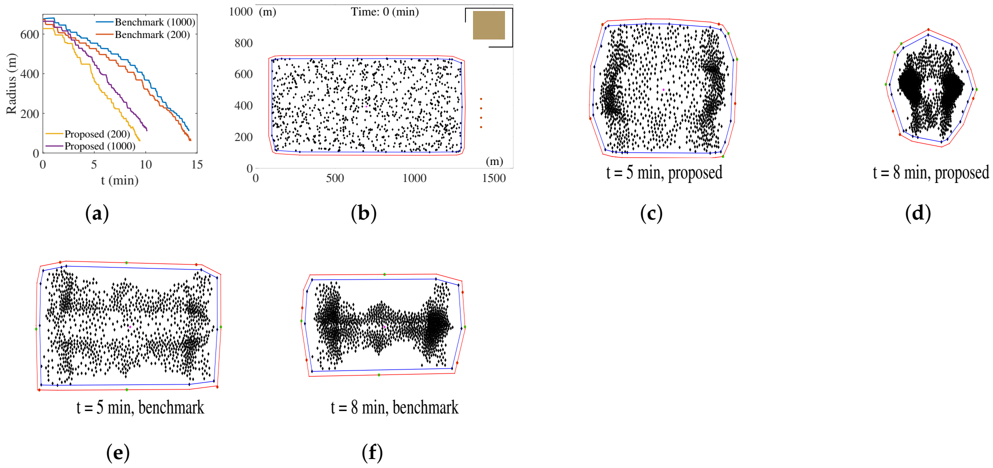
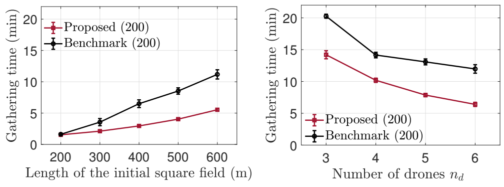


Figure 10. (**a**) Animals' footprint radius versus time t for herding 200 and 1000 animals with 4 barking drones using both the proposed and the benchmark method; (**b**) Initial distribution of the 1000 animals; (**c**–**f**) snapshots of the 1000 animals at $t = 5, 8$ min for both methods.

Interestingly, Figure 10a reveals that the difference between the time of gathering 200 animals and 1000 animals is not that obvious, with both the proposed method and the benchmark method. The proposed method, however, can always use less time to complete the gathering mission. This is because that the proposed method always chases and repulses the animals that are farthest to the centre, while the benchmark method is repulsing the animals indiscriminately. Therefore, the animals' footprint with the proposed method becomes increasingly round-like during shrinking, while animals' footprint with the benchmark method becomes long and narrow. This fact can be observed by comparing Figure 10c,e, and Figure 10d,f.

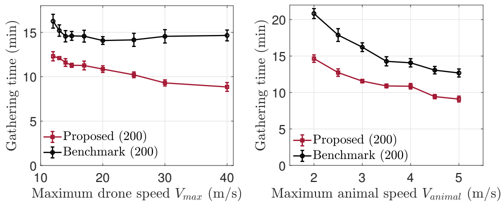
Note that, the time consumption of flying to the edge and the driving task mainly depend on the initial locations of the drones and the animals. From now on, we focus on evaluating the average gathering time after the drones have arrived on the extended hull. The aforementioned minor difference between herding 200 and 1000 animals is very likely because that the gathering time is strongly correlated with the size of the initial field, rather than the number of the animals. To confirm this, we change the initial field into square and investigate the relationship between the gathering time and the length of the initial square field; see Figure 11a. It reveals that the average gathering time increases significantly with the initial square field length. This supports the guess that the gathering time is strongly correlated with the size of the initial field. The reason is also that the gathering time mainly depends on the movement of the animals on the edge, and particularly the travelling time for them to move to the area close to C_o. With fixed V_{animal} and the same repulsion from the barking drones, the travelling distances of these animals are dominated by the size of the initial field. Moreover, Figure 11a shows that the difference between the gathering time of the benchmark method and the proposed method increases with the initial square field length. It means the benchmark method is more 'sensitive' to the varying length of the initial square field. We further investigate the relationship between the gathering time and the the number of barking drones n_d; see Figure 11b. Not surprisingly, the average gathering time decreases significantly with the increase of n_d for both methods. Besides, Figure 11b shows that the superiority of the proposed method becomes more obvious with n_d increases when $n_d \geq 4$.



(**a**) (**b**)

Figure 11. Comparisons of the average gathering time for different values of (**a**) length of the initial square field; (**b**) number of barking drones n_d.

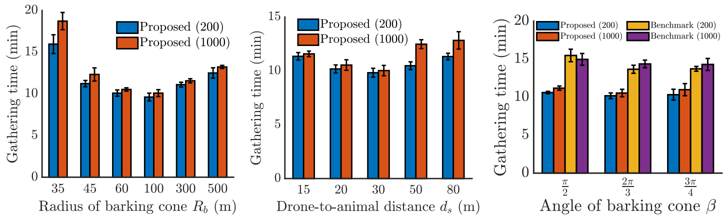
Next, we investigate the impact of the drone speed and animal speed on the gathering time; see Figure 12. Figure 12a shows that slower drones will lead to a higher average gathering time, especially when the maximum drone speed $V_{max} < 15$ m/s, for both the benchmark method and the proposed method. Moreover, the average gathering time of the benchmark method is more 'sensitive' to V_{max} when $V_{max} < 15$ m/s. In addition, in our simulations, drones with $V_{max} \leq 10$ m/s cannot accomplish the gathering task using the benchmark method. In the implementation of the proposed method, drones with $V_{max} > 15$ m/s is preferable. Furthermore, the average gathering time of the proposed method reduces with V_{max} increases, the percentage of the reduction, however, is not significant when $V_{max} > 30$ m/s. Figure 12b shows that animals with higher maximum speed V_{animal} can be gathered in a shorter time. Particularly, with the proposed method, the average gathering time reduces around 38% (from 14.7 min to 9.1 min) when V_{animal} increases 150% (from 2 m/s to 5 m/s). For the benchmark method, the average gathering time reduces around 39% (from 20.8 min to 12.7 min). Therefore, the reduction of average gathering time is much slower than the increase of V_{animal} when 2 m/s $\leq V_{animal} \leq 5$ m/s, for both methods.



(**a**) (**b**)

Figure 12. Comparisons of the average gathering time for different values of (**a**) the maximum drone speed V_{max}; (**b**) the maximum animal speed V_{animal}.

We investigate impact of the barking cone radius R_b, the drone-to-animal distance d_s, and angle of barking cone β on the gathering time; see Figure 13. Figure 13a presents the relationship between the barking radius R_b and the gathering time with 200 animals and 1000 animals, respectively. We can observe that increasing R_b will accelerate the gathering when $R_b \leq 100$ m. But when $R_b > 100$ m, the average gathering time increases with R_b, which is contradictory to our expectation. One possible reason is that the gathering time mainly depends on the animals on the edges. If R_b is too lager, it may cause the mutual interference between the repulsive forces inflicted by the barking drones, which may pull down the gathering. Moreover, Figure 13a shows that the proposed method is more 'sensitive' to R_b when gathering more animals with $R_b \leq 100$ m. This is because that the proportion of the repulsed animals near the edges tend to be less when n_s increases, for a fixed R_b.



(**a**) (**b**) (**c**)

Figure 13. Comparisons of the average gathering time for different values of (**a**) barking cone radius R_b; (**b**) drone-to-animal distance d_s; (**c**) angle of barking cone β.

Figure 13b suggests that the average gathering time decreases with d_s increases when $d_s \leq 30$ m. One possible reason is that more animals will be repulsed to the directions that are not point to the centre if d_s is too small, since the repulsive force from the barking drone is point to the opposite of it and the barking zone is fan-shaped. This result is also considered the interference. However, increasing d_s will decelerate the gathering when $d_s > 30$ m, and this becomes more obvious with more animals. It is reasonable because increasing d_s is almost equivalent to decreasing R_b when R_b is fixed.

Figure 13c indicates that the gathering time of the proposed method is significantly less than the one of the benchmark method among $\beta = \pi/2, 2\pi/3$ and $3\pi/4$ with 200 animals and 1000 animals. It can also be seen from Figure 13c that the gathering time does not show monotonicity with the increase of β. The gathering time with $\beta = 2\pi/3$, however, is slightly less than the cases with $\beta = \pi/2$ and $3\pi/4$, for both the proposed method and the benchmark method.

We are also interested in the impacts of the measurement errors from the 'observer' on our method. We add random noise to the measured positions of the animals, and the amplitude of the noise is from 2 to 10 m. We conduct 20 simulations with 200 animals and 1000 animals independently for each value with $d_s = 20$ m. The results are as shown in Figure 12. Under the measurement error, the average impact on the gathering time is relatively small. Figure 12 shows that the average gathering time increased slightly with the measurement error increases from 2 to 10 m for both cases. For example, from no measurement error to 10 m error, the average gathering time for 1000 animals increased from 11.7 min to 12.6 min. The difference is less than 1 min. Moreover, the impact of measurement errors on the average gathering time is less significant in the case of 200 animals (see Figure 14).

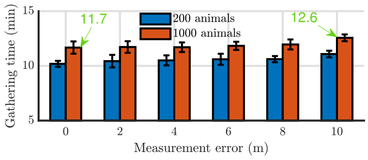


Figure 14. Impact of measurement errors.

In summary, we present computer simulation results in this section to demonstrate the performance of the proposed method. These results confirm that the proposed method can efficiently herd a large number of farm animals and outperform the benchmark method. By investigating the impact of the system parameters we can obtain that higher speed of drones leads to shorter gathering time. The barking cone radius R_b and the drone-to-animal distance d_s also significantly affect the gathering time. The optimal values of them can be obtained via experiments on real animals.

7. Conclusions

In this paper, we proposed a novel robotic herding system based on autonomous barking drones. We developed a collision-free sliding mode based motion control algorithm, which navigates a network of barking drones to efficiently collect a flock of animals when they are too dispersed and drive them to a designated location. Simulations using a dynamic model of animal flocking based on Reynolds' rules showed the proposed drone herding system can efficiently herd a thousand of animals with several drones. A unique contribution of this paper is the proposal of the first prototype of herding a large flock of farm animals by autonomous drones. The future work is to conduct experiments on real farm animals to test the proposed method. Moreover, the sound from drones may have some unknown effect on the animals and their responses. The study of such an aspect can also be carried out in field experiments.

Author Contributions: Conceptualization, X.L.; methodology, X.L.; software, X.L.; validation, X.L.; formal analysis, X.L.; investigation, X.L. and H.H.; resources, X.L.; data curation, X.L. and J.Z.; writing—original draft preparation, X.L.; writing—review and editing, X.L., H.H. and J.Z.; visualization, X.L.; supervision, A.V.S.; project administration, J.Z.; funding acquisition, A.V.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Australian Research Council. Also, this work received funding from the Australian Government, via grant AUSMURIB000001 associated with ONR MURI grant N00014-19-1-2571.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

References

1. Elijah, O.; Rahman, T.A.; Orikumhi, I.; Leow, C.Y.; Hindia, M.N. An overview of Internet of Things (IoT) and data analytics in agriculture: Benefits and challenges. *IEEE Internet Things J.* **2018**, *5*, 3758–3773. [CrossRef]
2. Birrell, S.; Hughes, J.; Cai, J.Y.; Iida, F. A field-tested robotic harvesting system for iceberg lettuce. *J. Field Robot.* **2020**, *37*, 225–245. [CrossRef] [PubMed]
3. Ahmed, N.; De, D.; Hussain, I. Internet of Things (IoT) for smart precision agriculture and farming in rural areas. *IEEE Internet Things J.* **2018**, *5*, 4890–4899. [CrossRef]
4. Marini, D.; Llewellyn, R.; Belson, S.; Lee, C. Controlling within-field sheep movement using virtual fencing. *Animals* **2018**, *8*, 31. [CrossRef] [PubMed]
5. Yao, Y.; Sun, Y.; Phillips, C.; Cao, Y. Movement-aware relay selection for delay-tolerant information dissemination in wildlife tracking and monitoring applications. *IEEE Internet Things J.* **2018**, *5*, 3079–3090. [CrossRef]
6. Achour, B.; Belkadi, M.; Aoudjit, R.; Laghrouche, M. Unsupervised automated monitoring of dairy cows' behavior based on Inertial Measurement Unit attached to their back. *Comput. Electron. Agric.* **2019**, *167*, 105068. [CrossRef]
7. Vaughan, R.; Sumpter, N.; Frost, A.; Cameron, S. Robot Sheepdog Project Achieves Automatic Flock Control. In *Proceedings of the Fifth International Conference on the Simulation of Adaptive Behaviour*; MIT Press: Cambridge, MA, USA, 1998; pp. 489–493. Available online: https://ieeexplore.ieee.org/document/6278703 (accessed on 28 May 2020).
8. Sumpter, N.; Bulpitt, A.J.; Vaughan, R.T.; Tillett, R.D.; Boyle, R.D. *Learning Models of Animal Behaviour for a Robotic Sheepdog*; MVA: Chiba, Japan, 1998; pp. 577–580.
9. Evered, M.; Burling, P.; Trotter, M. An investigation of predator response in robotic herding of sheep. *Int. Proc. Chem. Biol. Environ. Eng.* **2014**, *63*, 49–54.
10. BBC. Robot Used to Round Up Cows is a Hit with Farmers. Available online: https://www.bbc.com/news/technology-24955943 (accessed on 28 May 2020).
11. Sciencealert. Spot the Robot Sheep Dog. Available online: https://www.sciencealert.com/spot-the-robot-dog-is-now-herding-sheep-in-new-zealand (accessed on 28 May 2020).
12. IEEE Spectrum. Swagbot to Herd Cattle. Available online: https://spectrum.ieee.org/automaton/robotics/industrial-robots/swagbot-to-herd-cattle-on-australian-ranches (accessed on 28 May 2020).
13. Telegraph. Britains Most Expensive Sheepdog. Available online: https://www.telegraph.co.uk/news/2016/05/14/britains-most-expensive-sheepdog-sells-for-15000-at-auction/ (accessed on 28 May 2020).
14. Gazi, V.; Fidan, B.; Marques, L.; Ordonez, R.; Kececi, E.; Ceccarelli, M. Robot swarms: Dynamics and control. In *Mobile Robots for Dynamic Environments*; eBooks; ASME: New York, NY, USA, 2015; pp. 79–107.
15. Strömbom, D.; Mann, R.P.; Wilson, A.M.; Hailes, S.; Morton, A.J.; Sumpter, D.J.; King, A.J. Solving the shepherding problem: Heuristics for herding autonomous, interacting agents. *J. R. Soc. Interface* **2014**, *11*, 20140719. [CrossRef] [PubMed]
16. Hoshi, H.; Iimura, I.; Nakayama, S.; Moriyama, Y.; Ishibashi, K. Robustness of Herding Algorithm with a Single Shepherd Regarding Agents' Moving Speeds. *J. Signal Process.* **2018**, *22*, 327–335. [CrossRef]

17. Hoshi, H.; Iimura, I.; Nakayama, S.; Moriyama, Y.; Ishibashi, K. Computer simulation based robustness comparison regarding agents' moving-speeds in two-and three-dimensional herding algorithms. In Proceedings of the 2018 Joint 10th International Conference on Soft Computing and Intelligent Systems (SCIS) and 19th International Symposium on Advanced Intelligent Systems (ISIS), Toyama, Japan, 5–8 December 2018; pp. 1307–1314.
18. Pierson, A.; Schwager, M. Bio-inspired non-cooperative multi-robot herding. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 1843–1849.
19. Pierson, A.; Schwager, M. Controlling noncooperative herds with robotic herders. *IEEE Trans. Robot.* **2017**, *34*, 517–525. [CrossRef]
20. Singh, H.; Campbell, B.; Elsayed, S.; Perry, A.; Hunjet, R.; Abbass, H. Modulation of Force Vectors for Effective Shepherding of a Swarm: A Bi-Objective Approach. In Proceedings of the 2019 IEEE Congress on Evolutionary Computation (CEC), Wellington, New Zealand, 10–13 June 2019; pp. 2941–2948.
21. Vayssade, J.A.; Arquet, R.; Bonneau, M. Automatic activity tracking of goats using drone camera. *Comput. Electron. Agric.* **2019**, *162*, 767–772. [CrossRef]
22. Barbedo, J.G.A.; Koenigkan, L.V.; Santos, P.M.; Ribeiro, A.R.B. Counting Cattle in UAV Images' Dealing with Clustered Animals and Animal/Background Contrast Changes. *Sensors* **2020**, *20*, 2126. [CrossRef] [PubMed]
23. Huang, H.; Savkin, A.V. An Algorithm of Reactive Collision Free 3-D Deployment of Networked Unmanned Aerial Vehicles for Surveillance and Monitoring. *IEEE Trans. Ind. Inform.* **2020**, *16*, 132–140. [CrossRef]
24. Li, X.; Huang, H.; Savkin, A.V. A Novel Method for Protecting Swimmers and Surfers From Shark Attacks Using Communicating Autonomous Drones. *IEEE Internet Things J.* **2020**, *7*, 9884–9894. [CrossRef]
25. Huang, H.; Savkin, A.V. A Method for Optimized Deployment of Unmanned Aerial Vehicles for Maximum Coverage and Minimum Interference in Cellular Networks. *IEEE Trans. Ind. Inform.* **2019**, *15*, 2638–2647. [CrossRef]
26. Savkin, A.V.; Huang, H. Navigation of a Network of Aerial Drones for Monitoring a Frontier of a Moving Environmental Disaster Area. *IEEE Syst. J.* **2020**, *14*, 4746–4749. [CrossRef]
27. Paranjape, A.A.; Chung, S.J.; Kim, K.; Shim, D.H. Robotic herding of a flock of birds using an unmanned aerial vehicle. *IEEE Trans. Robot.* **2018**, *34*, 901–915. [CrossRef]
28. RaisingSheep. Sheep Herding Dogs. Available online: https://http://www.raisingsheep.net/sheep-herding-dogs.html/ (accessed on 28 May 2020).
29. The Washington Post. New Zealand Farmers Have a New Tool for Herding Sheep: Drones that Bark Like Dogs. Available online: https://www.washingtonpost.com/technology/2019/03/07/new-zealand-farmers-have-new-tool-herding-sheep-drones-that-bark-like-dogs/ (accessed on 28 May 2020).
30. The Wall Street Journal. They're Using Drones to Herd Sheep. Available online: https://www.wsj.com/articles/theyre-using-drones-to-herd-sheep-1428441684 (accessed on 28 May 2020).
31. Li, X. Some Problems of Deployment and Navigation of Civilian Aerial Drones. *arXiv* **2021**, arXiv:2106.13162.
32. Chengzhi Drone. MP130 Drone Digital Voice Broadcasting System. Available online: https://www.gzczzn.com/productArgumentsServlet?productId=MP130/ (accessed on 28 May 2020).
33. DJI. Matrice 300 RTK. Available online: Https://www.dji.com/au/matrice-300 (accessed on 28 May 2020).
34. AIROBOTICS. Automated Industrial Drones. Available online: Https://www.airoboticsdrones.com/ (accessed on 28 May 2020).
35. Sun, Y.; Xu, D.; Ng, D.W.K.; Dai, L.; Schober, R. Optimal 3D-trajectory design and resource allocation for solar-powered UAV communication systems. *IEEE Trans. Commun.* **2019**, *67*, 4281–4298. [CrossRef]
36. Fujioka, K.; Hayashi, S. Effective shepherding behaviours using multi-agent systems. In Proceedings of the 2016 IEEE Region 10 Conference (TENCON), Singapore, 22–25 November 2016; pp. 3179–3182.
37. Reynolds, C.W. Flocks, herds and schools: A distributed behavioral model. In Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, Anaheim, CA, USA, 27–31 July 1987; pp. 25–34.
38. Wang, C.; Savkin, A.V.; Garratt, M. A strategy for safe 3D navigation of non-holonomic robots among moving obstacles. *Robotica* **2018**, *36*, 275–297. [CrossRef]
39. Utkin, V.I. *Sliding Modes in Control and Optimization*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013.
40. Drakunov, S.V.; Utkin, V.I. Sliding mode control in dynamic systems. *Int. J. Control* **1992**, *55*, 1029–1037. [CrossRef]
41. Savkin, A.V.; Evans, R.J. *Hybrid Dynamical Systems: Controller and Sensor Switching Problems*; Birkhauser: Boston, MA, USA, 2002.
42. Fujioka, K. Effective herding in shepherding problem in v-formation control. *Trans. Inst. Syst. Control Inf. Eng.* **2018**, *31*, 21–27. [CrossRef]