2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)
Bari, Italy. October 6-9, 2019

Leader And Predator Based Swarm Steering For Multiple Tasks

Raghavv Goel, John Lewis, M.A. Goodrich and P.B. Sujit

***Abstract*— A robotic swarm can perform various tasks. However, a human is required to task the swarm. Human control over the swarm can be enabled through a set of influential agents which can be either leaders or predators. In the presence of multiple tasks, the swarm may need to split into sub-swarms to accomplish the task and re-group as a swarm to execute larger tasks. The response of the swarm in the presence of influential agents depends on the swarm dynamics. A precise measure of influence using leaders or predators or a combination of leaders and predators to achieve the mission is not adequately studied. In this paper, we analyze the effect of using only leaders, only predators and a combination of leaders and predators on three swarm models namely, shepherding model, Couzin's model and a physicomimetic models while they perform foraging tasks and carry out Monte-Carlo simulations to evaluate the performance of the influential agents on different swarms. We also propose a novel way to split a swarm into smaller sub-swarms using influential agents. Our results show that the predator based swarm splitting and steering to a task based on shepherding model performs far better than any other combination of leaders and predators. This result is consistent even when the number of agents is increased to 500.**

I. Introduction

A swarm of robots can perform several tasks simultaneously and provide robustness to the mission [1], [2], [3]. Nature inspired swarm robotic behaviors like the school of fish [4], swarm of birds [5], [6], migration[7], etc., can be modelled using simple rules of interaction between individuals [5], [8]. The rules are decentralized, computationally cheap, and hence encoding them into the robotic swarm is simple and can scale to large numbers of agents. These behaviors can be exploited to create sub-swarms for simultaneous execution of several tasks or join them to execute a complex task.

A common task involving a robotic swarm is to navigate from the current location to another (task location) as shown in Figure 1 and perform some action. The task could be cleaning or acquire information at the task location or performing some other action. In order to perform the task, agents must have some higher level control that provides information about the task. There are several ways of imparting this higher knowledge to the agents [9]. One way is to broadcast the task location to the agents. This approach requires significant bandwidth for large-scale deployments which may limit their use in remote applications. Another approach to control large-scale robotic swarm is by introducing a few influential agents, which can manipulate the swarm to steer them towards the goal [10]. These influential agents can be leaders that have an attraction property or predators that have a repulsion property.

Simulated swarms of robots can be controlled by a few leader agents [11], who can change their role to allow a broad range of group patterns to emerge [12], can change their region of influence and speed [13]. Leaders can be chosen/used by dynamically selecting agents based on their connectivity within the swarm network [14] or strategically placing them in the swarm [15], [16]. Although, leader-based steering is simple to implement and enables steering due to the attraction property, leader agents do not have the responsibility or role in ensuring that the complete swarm has reached the goal. In fact, in leader-based swarm steering, some agents at the back of the swarm may be lost [17], [15].

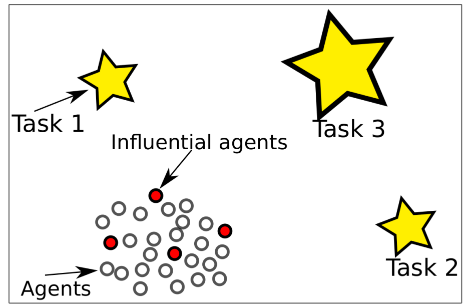


Fig. 1: The robot swarm needs to be steered to the task location region using few influential agents (either leaders or predators or a combination of leaders and predators) to perform the tasks of varying size.

Contrary to the leader-based swarm control, predator-based swarm control potentially enables the transfer of all the agents to the goal without losing any [18], [19]. Typically, predator-based steering is called shepherding which has received attention over several years. Initial experiments on steering a group of ducks using a robot shows that the predator based steering has potential [20]. Several types of predators based models have been used for multi-robot systems. A rule-based mechanism to steer a swarm through shepherding was developed in [21], [22], [23]. One way to automate the process of shepherding behavior is to learn the behavior. A reinforcement learning algorithm using SARSA was proposed in [24] for a single predator. However, predators are more difficult to control compared to their leader-based alternative [25]. Also, different swarm models react differently to the presence of a predator [10]. For instance, if we consider the Couzin's model [8] in the swarm state, then the swarm may disintegrate in the presence of a predator. On the other hand, if we consider the shepherding model [19], then the swarm becomes cohesive in the presence of a predator. Another approach of using predators to steer a flock is to cage them [26] such that all the agents are inside the circle formed by the predators. In this case, as the flock size increases, the number of predators required to steer the swarm may also increase.

In a mission, there may be several tasks, and hence the swarm may be split into multiple sub-swarms to accomplish various tasks simultaneously and in some cases join few sub-swarms to create a bigger swarm to accomplish larger tasks [27]. The above articles primarily focus on a single task. The swarms can be split into sub-swarms by broadcasting commands to the agents directly [28], by generating an imaginary line and the agents can determine which side of the line they belong to form a sub-group [29], using artificial potential fields to split the swarm into two groups [30], [31], or by selecting the sub-swarm agents while maintaining the sub-swarm network connectivity [27].

In the literature, most of the works focus on steering a swarm either using leader-based or predator-based mechanism for single tasks. There is no adequate study on the performance of the leader-based and predator-based influence on the same swarm and also how the performance varies between splitting the swarm into sub-swarm using leader or predator based models. This is essential as it may allow the human operator to switch between leader and predator based steering mechanisms to accomplish different tasks simultaneously while minimizing the interaction effort of the human operator by delegating the responsibility to these influential agents. In this paper, we study the influence of leaders and predators on three different type of swarm models, namely, a shepherding model [19], Couzin's model [8] and a physicomimetics model [32]. We select these three models because of different underlying principles for swarm behaviour and these swarm models encapsulate most of the leader and predator based models in the literature. We evaluate the performance of these models under the time to accomplish multiple tasks and the number of agents lost metrics. We further evaluate the splitting ratio of the agents, effect of swarm size, and increase in number of influencing agents.

II. Swarm Models

Swarms can be influenced either by using predators or leaders to steer them towards the goal. We will now briefly describe these models.

A. Shepherding model

We use the shepherding model given in [19], as this model has been validated experimentally on real world sheep and sheep-dog. In this model, the motion of sheep as a swarm, is governed by five forces. Consider a scenario as shown in Figure 2a, where an influenced agent (green circle) is affected due to the presence of a predator(blue square). Due to the presence of the predator, a force F acts on the agent(located at $\bar{S}_i$) to move away from the predator(located at $\bar{P}_j$) which can be calculated using Equation (1)), a force C acts on the agent to move towards its neighbors which can be evaluated using Equation (2)). There is always an inter-agent repulsion(denoted by vector R_e in Figure 2a and Equation (3)) when the sheep comes within a threshold(denoted by r_a) of another sheep . Also, there is a tendency for the agent to continue in a previously moved direction due to inertia (Equation (4)) and finally, a small error in a random direction (Equation (5)). The resultant of all these forces decide the direction of the sheep's next step. Therefore, from these equations of motion, we can find that the sheep become cohesive in the presence of a shepherd (predator), otherwise they disperse and roam freely. The various parameters involved in the formulation is given in Table I.

The notations for the below equations are present in Table I.

$$F = -\rho_s \frac{(\bar{P}_j - \bar{S}_i)}{||\bar{P}_j - \bar{S}_i||} \quad , \forall P_i \in \mathcal{P} \tag{1}$$

$$C = \frac{c}{n}\sum_{j=1}^{n} \bar{S}_j \tag{2}$$

$$R_e = \begin{cases} \rho_a \dfrac{(\bar{S}_i - \bar{S}_j)}{||\bar{S}_i - \bar{S}_j||} & , \forall |\bar{S}_i - \bar{S}_j| \le r_a, \\ 0 & \text{, otherwise,} \end{cases} \tag{3}$$

$$I = h\dot{\bar{S}}_i \tag{4}$$

$$S_e = p * e * v_s \quad \forall S_i \in S_g \tag{5}$$

B. Couzin's model

Couzin's model basically describes the collective behavior of fish which is more dynamic than the sheep swarm model. The model is defined by three different zones, namely, repulsion zone (RR), orientation zone (RO), and attraction zone (RA), as shown in Figure 2b. The agents within the zone of repulsion (RR) repel each other which is given by the vector F_R. The agents in the zone of orientation (RO) orient along the average direction of all the neighbors (vector F_O). The agents that do not satisfy the above zone but are within the zone of attraction (RA) will be attracted towards each other which is given by (vector F_A). The agents under repulsion will not undergo orientation or attraction. Otherwise, the average of the resultant vector of attraction and orientation is calculated. The distance between two agents i and j is r_{ij} and the normalized direction of motion of the agent i is v_i. Then the swarm can be mathematically modelled with Equations (6), Equations (7) and (8).

$$F_R = -\sum_i \frac{r_{ij}}{|r_{ij}|}, \quad \forall i, i \in RR \tag{6}$$

$$F_O = \sum_i \frac{v_i}{|v_i|}, \quad \forall i, i \in RO \tag{7}$$

$$F_A = \sum_i \frac{r_{ij}}{|r_{ij}|} \quad \forall i, i \in RA. \tag{8}$$

Tuning these radii will result in various swarm behaviours – randomly moving swarm with almost stationary centroid, torus, parallel moving swarm and highly parallel swarm. We consider the highly parallel swarm mode which can be attained with a high RO compared to RR. This mode provides a handle to influence the swarm through predators and leaders.

TABLE I: Sheep Parameters

Parameter	Description	Parameter	Description
h	Relative strength of inertial effect	R_e	Inter-agent repulsive vector
n	Number of nearest neighbours	I	Inertial vector
C_M	Centre of mass of flock	C	Clustering vector
S_l	Furthest sheep position	F	Predatory vector
v_s	Speed of sheep	p	Probability of random movement
e	Strength of angular noise	c	Strength of clustering effect
ρ_s	Strength of predatory effect	ρ_a	Relative strength of inter-agent repulsion

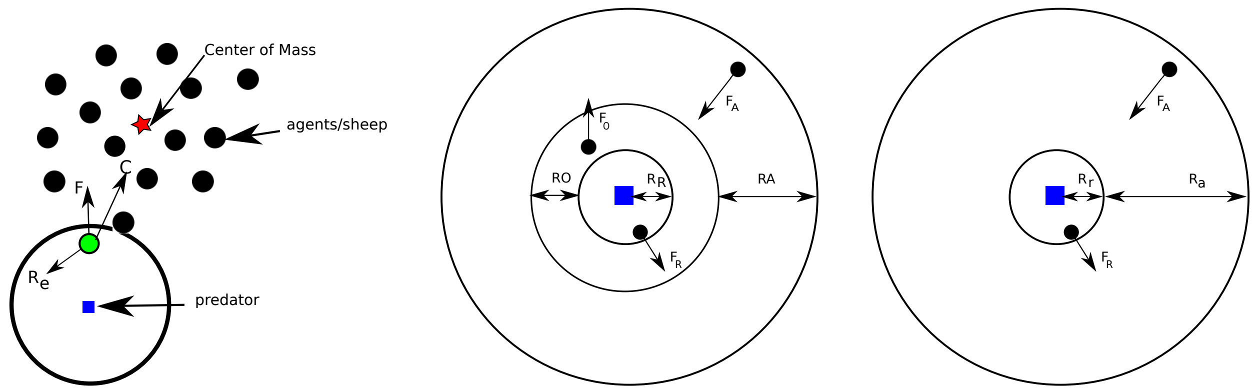


Fig. 2: (a) Forces acting on an agent (green colored) under a predator influence for a shepherding model (b) Radii and Forces involved in the Couzin's model. (c) Forces involved in the Physicomimetic model

C. Physicomimetic model

$$F = ma = m\frac{\Delta V}{\Delta t}, \tag{9}$$

$$F = c\frac{Gm_im_j}{r_{ij}^2}, \quad \forall\ i \in \mathcal{P} \tag{10}$$

$$c = \begin{cases} 1 & ,\forall\ i \in R_A, \\ -1 & ,\forall\ i \in R_R, \\ 0 & ,\text{otherwise.} \end{cases}$$

Physicomimetics is a physics based model and hence this model is stationary when there is no external influence [32]. In this model, each agent is a particle that experiences a gravitational force and forces induced by other agents within a finite neighborhood. The model is defined by two zones – repulsion zone (R_r) and zone of attraction (R_a) as shown in Figure 2c. All the agents within R_r repel each other, while the other agents that are within R_a tend to move towards each other. The force F, is defined by the Newton's Law of Gravitation, which is dependent on the mass of the particles and the inverse of the square of the distance between the particles as shown in Equation (10). In order to limit the momentum of the particles, the maximum force that can be attained is F_{max}. Similarly, the maximum velocity is capped at V_{max}. For an agent i, the forces are given by Equation (10) and the velocity is calculated using Equation (9). The mass m_i and m_j represent the mass of the agents which is assumed to be 1.

III. Influencing Agent Models

The swarm can be influenced using leaders and predators. We will now describe, how these leaders and predators are modelled.

A. Predator model

The predator model utilizes the repulsion mechanism to influence the swarm. As the swarm size increases, it is difficult for a single predator to steer the swarm and hence multiple predators are required. Initially, we will describe the single predator mechanism and then extend the model to multiple predators for different swarm models.

1) Single predator : For the shepherding model, the predator attempts to collect the agents that are straying away from the cluster by positioning itself at the collecting point, P_c. P_c is calculated in Equation (11) and is given by a unit vector directed away from C_M (Center of mass) at S_l (position of the sheep furthest from C_M), scaled by a constant c_c. Due to predatory effect, a stray agent moves in the opposite direction of the predator location with some bias towards the rest of the swarm members. Once collected, the agents are collectively herded towards the goal. The shepherd carries this out by positioning itself at the driving point, P_d. P_d explained in Equation (12) and is modelled as the unit vector directed away from D (the destination or goal) at C_M, scaled at a constant c_d. Herding of the sheep can be visualized in Figure.2a.

The switch between collecting and herding plays a vital role in the steering of the flock. A sheep needs to be collected if it strays far away from the flock. This is modelled by calculating $||S_l - C_M||$. If this distance is greater than a predefined threshold, T, then the shepherd attempts to collect the straying sheep else it herds the entire flock (summarized in (13)). This handshaking model is similar to the shepherding model presented in [19].

$$P_c = S_l + c_c \frac{S_l - C_M}{||S_l - C_M||} \tag{11}$$

$$P_d = C_M + c_d \frac{C_M - D}{||C_M - D||} \tag{12}$$

$$mode = \begin{cases} herding & \text{if } ||S_l - C_M|| < T \\ collecting & \text{if } ||S_l - C_M|| > T \end{cases} \tag{13}$$

In case of Couzin's highly parallel model, we can take advantage of the orientation of the swarm to ease the entire process. The predator places itself behind the swarm in the direction of the goal, thereby propelling a part of the herd towards the goal. This orientation towards the goal will eventually spread through the entire herd. Once the swarm's heading angle is directed towards the goal, the predator continuously monitors to ensure that all the agents are directed towards the goal. In a situation where an agent heads away from the goal, the predator corrects the agents heading angle by placing itself behind the straying agent, facing the goal.

Unlike the sheep model, the physicomimetic model does not have the clustering effect. Thus, it is difficult to control and influence the herd at the same time. Hence we require a handshake model of herding and recollecting. The model will recollect itself as long as the agents are all within the radius of attraction of each other. Thus the shepherds are designed to herd until the swarm elongates beyond a threshold. After this, the predator moves away to allow the herd to recollect itself. This repeated process is performed until the swarm reaches the goal.

2) Multiple predators: The multi-predator technique is applied by partitioning the swarm into N_p sections, where N_p is the number of predators involved. Each section is then monitored and manipulated by the predators assigned to their respective sections. In order to improve the efficiency of the multi-predator technique, rather than having the entire swarm partitioned, we partition a section ($Q(0, 2\pi)$) which is further away from the goal. For multi-shepherd herding with N_p shepherds, the $Q-$ sectioned flock is partitioned into N_p parts. Then each sector is steered separately, following rules similar to Equations (12) and (11). Equation (14) calculates the driving position of the $i-th$ shepherd, $P-d(i)$ monitoring $i-th$ sector. The centre of mass of the sheep within the $i-th$ sector is given by $C_M(i)$. This reduces the number of agents to be monitored, hence making it easier for the shepherd to collect and herd. This is shown in Figure 3a. The angle made by the shepherds at the centre of mass is Q. The shepherds are positioned symmetrically such that the net force is towards the goal as shown in Figure 3a. Although this technique is similar to the caging technique in [26], the partitioning is performed only at the bottom half of the

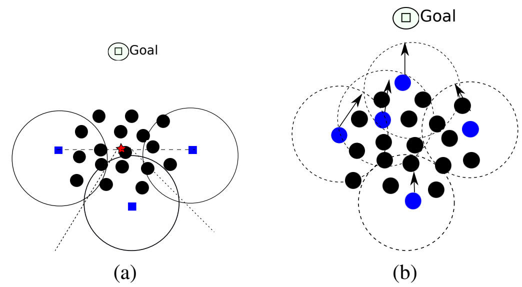


Fig. 3: (a) The single predator approach extended for multiple predators (b) Modified multi-leader approach of the sheep-dog to influence the sheep in leader mode

swarm. Also the predators are free to select between herding and collecting whenever the agents are straying away. Due to this, the number predators required for our approach is far less than than that of [26].

$$P_d(i) = C_M(i) + c_d \frac{C_M(i) - D}{||C_M(i) - D||} \; \forall \; i, i \in (1, N_p) \tag{14}$$

On similar lines, Equation (15) calculates the collecting position, $P_c(i)$, to retrieve a deviating sheep from the i_{th} sector, positioned at $S_l(i)$, where $S_l(i)$ depicts the agent straying away from the monitored sector i.

$$P_c(i) = S_l(i) + c_c \frac{S_l(i) - C_M(i)}{||S_l(i) - C_M(i)||} \; \forall \; i, i \in (1, N_p) \tag{15}$$

B. Leader model

As opposed to "*fear*" in the predator model, the agents have an affinity towards the leader. The agents within the range of the leader's influence are motivated to follow the leader. This idea of pursuing the leader will not impart the clustering in the sheep model due to the absence of fear. Thus we have replaced the clustering away from the shepherd with clustering towards the leader. While in a physicomimetic model, N_s leaders are chosen amongst the agents, after allowing them to acquire equilibrium. The leaders are not influenced by other agents but are motivated in reaching the goal as shown in Figure 3b. The affinity towards the leader propels the remaining agents to follow the leader towards the goal. We have used this idea of leader-based herding in the physicomimetic and shepherding models.

For Couzin's model, we have a similar approach. After the swarm acquires a steady state, N_s leaders are chosen randomly. However , the leaders in this model follow the rules of repulsion and is passive to the other forces involved. If the leaders are not under repulsion, they move towards the goal else, they repel each other. The other agents attempt to orient themselves with the leaders abiding by the three rules of spatial orientation [11]. Due to the high group orientation within the agents, they attempt to orient themselves with the leader and move towards the goal.

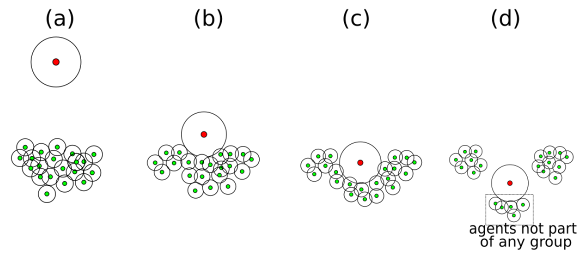


Fig. 4: Predator in red colour, sheep agents in green colour, the circular boundary around the sheep agent represents it's attraction radius while for the predator it is the predator sheep repulsion radius. A single predator trying to split the swarm but the result is a fragmented swarm.

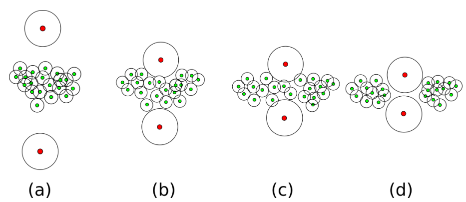


Fig. 5: When two predators moving towards the swarm in opposite directions effect the swarm to split uniformly.

IV. Splitting and joining of swarms

There are several ways to split a swarm as described in Section I. Instead of creating artificial barriers or artificial potential fields, we split the swarm by using the natural repulsion response of the swarm in the presence of a predator. For leaders, splitting the swarm is a difficult task due to their attraction property. When the leaders move in different directions, the leader neighbors may be attracted to their nearest leader and move along with the leader. However, when the leaders influence is not persistent then some of the agents may be left without any leaders. Due to this reason, we use only predators to split the swarm.

Using only a single predator for splitting the swarm is difficult and there is a possibility of swarm fragmentation as shown in Figure 4. Therefore, we introduce two predators moving towards the swarm from opposite directions as shown in Figure 5. The predators are positioned opposite to each other with the swarm between them. As the predators move towards each other, the agents are influenced by the repulsion response of the predators. As they repel, the swarm splits into two sub-swarms. Note that if the attraction property of the agents is too less then the whole swarm may be split into multiple sub-swarm. Hence, the attraction range of the agents is a design parameter which needs some attention.

The splitting process is scalable to any number of sub-groups but the number of agents per group may not be equal as the process is not selective. By positioning the predators in opposite directions at different angles, one can split the swam into different sizes as well. An illustration of splitting a swarm into four by using four predators is shown in [33].

Similarly, joining sub-swarms to create a bigger swarm

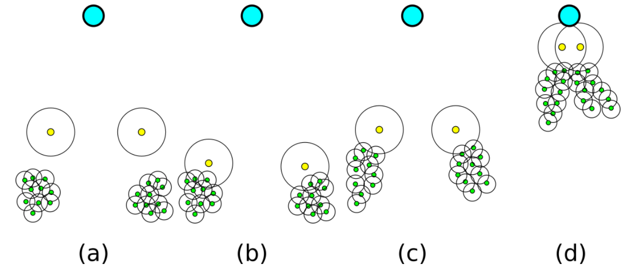


Fig. 6: Agents are green, the leaders are yellow and the cyan circle is the goal. The boundary of agents is the clustering radius of each agent. The boundary of leaders is the attraction radius of leaders towards agents.

can be performed. The influential agents are tasked to bring their respective swarms to a predefined task location where all the agents regroup themselves. Figure 6 shows leader based joining of the swarm. Similar to leader-based joining, one can use predators to join sub-swarms.

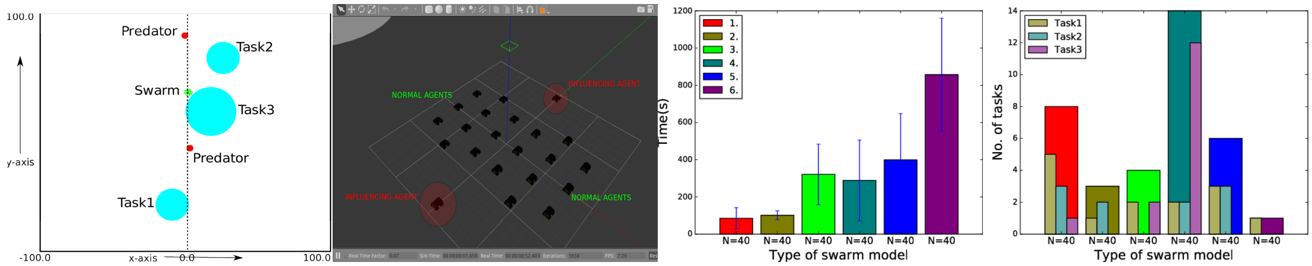
V. Results

We analyze the effect of leader and predator based steering of a swarm performing multiple tasks. Initially, we perform the simulations using holonomic robots and then show their performance with non-holonomic robots.

A. Simulation setting

We consider an environment of 200m $\times$ 200m area as shown in Figure 7a. Initially the agents are randomly distributed around $(0, 40)$ $(\mathcal{N}(0,1))$. Two tasks, $T1$ and $T2$ are randomly initialized. The task $T1$ is randomly generated between $x(-100, 0)$ and $y(-100, 100)$. The task $T2$ is randomly generated between $x(0, 100)$ and $y(-100, 100)$. Both these tasks are circular with a radius of 10m, a task is said to be completed when it's radius is reduced to below 0.1m which implies area is below 0.034. The rate at which the radius decreases is proportional to the number of agents present at the tasks, higher the number of agents, the faster the task is completed. Once the initial tasks, $T1$ and $T2$, are completed, a new task $T3$ is generated randomly, and is bigger than the previous tasks. It has a radius of 15m and with the added constraint that the radius/area will only start to decrease if agents from both the sub-swarms are present, that is both the sub-swarms must join. This setup will verify the ability of the swarm to split and rejoin. The task $T3$ is generated in the area with the only constraint that it should not overlap with $T1$ and $T2$. Each agent velocity is $1\ m/s$.

We carry out twenty five simulations for each setting using ROS and RVIZ as simulator for point mass simulations. We also used Gazebo for non-holonomic simulations using Husky ground robot that can mimic a real world scenario as shown in Figure 7b. We used ROS because it is very easy to port to the real world based experiments. We analyze the performance of leaders and predators based steering in accomplishing the tasks in terms of time taken to complete and the number of agents left in the area.



(a) (b) (c) (d)

Fig. 7: (a) The cyan circles are the tasks, two small tasks (Task 1 and Task 2) and one larger task (Task 3). The swarms needs to complete Task 1 and 2 before starting Task 3. (b) Gazebo simulator with 20 agents (in the center) having distance of 3m (on both sides and front and back) between them and 2 influencing agents(marked in red color) (c) Performance of various combination of leaders and predators to split and rejoin the swarm while executing the three tasks. (d) The number of incomplete tasks by the swarm and the number of tasks per configuration

B. Analysis

Each swarm model can be influenced by a leader, a predator or a combination of both. We will evaluate the following combinations,

1) Shepherding with predator based splitting and leader-based steering
2) Shepherding with predator based splitting and steering
3) Couzin swarm with predator based splitting and predator becoming the leader
4) Couzin swarm with predator based splitting and randomly selecting leaders
5) Physicomimetics swarm with predator based splitting and steering
6) Physicomimetics swarm with predator based splitting and randomly selecting leaders.

C. Average time taken to accomplish the tasks

Figure 7c shows the average performance in terms of the average time taken to accomplish the tasks with a time of 1200 seconds for various combination of influential agents. The swarm size is of 40 agents and one predator/leader is allocated for each sub-swarm. From the figure, we can see that the shepherding model with predator based splitting and using leaders to steer the sub-swarm performs better than the rest. However, the standard deviation of this swarm is higher than the shepherding swarm with predator based splitting and steering. Thus these two variations for the shepherding model preform far better than the Couzin and Physicomimetics models.

Couzin's model does not perform well in the case of predator based steering, because the group is in motion and steering them with a single predator is difficult. On the other hand, since there is only one leader, the leader influence is not sustaining. Similarly, in the case of physicomimetic swarm, the swarm is rigid and hence persistent leader/predator influence is required. Due to the low number of predators and leaders, they are unable to complete the task. Therefore, we increased the number of leaders for Couzin and Physicomimetic swarms to 10% of the total agents present, implying that as the number of agents are 40, the number of leaders are 4 (double the number of predators). The number of predators were kept constant for all the models for comparison purposes.

Figure 7d shows the number of tasks left after allowing 1200 seconds of the simulation. Please note that the colour of the thicker bars in figures 7c and 7d, refer to the same swarm type from the list of 6 swarms (Section V-B). From the figure, we can see that the Shepherding with predator and leaders has higher number of tasks left compared to the predator based steering and splitting. This is because, in the leader based steering, the leader is not responsible for all the agents to be in the swarm and hence some of the agents are lost. This results in the swarm being smaller and not being able to complete the task within the given period. The Couzin swarm with predator based splitting and randomly selecting the leader is the strategy that performs the worst. Although the physicomimetics swarm accomplishes most of the tasks, the average time taken is very high.

Figure 8a shows the number of agents lost during the execution of the mission. From the figure we can see that shepherding model with predator based splitting and steering performs the best with no agents lost. This is intuitive because of the sheep model which influences the agent to go towards the center of the sheep mass. Similar to the shepherding model, the physicomimetics model also performs well in minimizing the number of agents lost. However, due to the leader based steering in shepherding and Couzin based swarming several agents are lost.

D. Splitting ratio

All the models use predator for splitting the swarm. Once the swarm is split, the ratio of the number of agents in the sub-swarm after splitting has an affect on the performance. A splitting ratio of 1 means perfect split between the total number of agents and a ratio of 0 means the predator failed to split the swarm. From Figure 8b we can see that the average splitting ratio of physicomimetics and shepherding models are almost similar around 0.8 for varying number of agents. However, the Couzin swarm has very low ratio because of the motion of the swarm while the predators are influencing it and also the speed at which the predator is intercepting the swarm. On the other hand, shepherding and physicomimetic swarms have low group motion compared to the Couzin form.

When we increase the speed of the predator to 0.35m/s (see Figure 8c), then we can see that the Couzin swarm model's splitting ratio has increased. This shows that due to the dynamic nature of this swarm, we need a predator with higher speed to have balanced sub-groups. On the other hand, with increased predator speed for a given swarm size, the ratio decreases for the shepherding swarm.

E. Comparison between splitting the swarm and no splitting

From the above results, the shepherding model performed far better than the other two models and hence we analyzed the effect of splitting the swarm and no splitting for the shepherding swarm. Figure 9a shows the effect of splitting the swarm into sub-swarm and using the complete swarm to accomplish the tasks sequentially. From the figure, we can see that when we split the swarm into sub-swarms, on average the performance is far better than no-splitting. The performance of the swarm splitting is significantly better when we increase the number of agents to 500 as shown in Figure 9b. This shows that splitting the swarm is scalable.

F. Effect of increasing the number of influential agents

One of the advantages of using the predator based steering is that the predators need to influence on only one side of the swarm while allowing the the swarm to move in the task direction. Thus, the area that needs to be influenced for the swarm is limited. Therefore, if we increase the number of predators, then the area that needs to be influenced by each predator reduces. Therefore, with increase in more predators, the performance of steering does not improve. This hypothesis can be seen from the results for 40 agents with increasing number of predators in Figure 9c.

G. Simulation using non-holonomic ground rovers

We used 20 agents (Huskies) and 2 influencing agents as shown in Figure 7b to split and steer shepherding swarm and Couzin swarm. These robots have heading angle and velocity constraints. For Couzin model one leader each was chosen from the sub-groups. The tasks T1 (-30,0) and T2 (30,0) were 30m away from the origin, and the final task T3 was at (0,-10). T3 is executed only when agents from both the sub-swarms are present. The speed of agents is 0.25m/s. The shepherds speed is 1.25m/s during splitting the swarm and 0.75m/s after the splitting. These speed values were selected after simulating the system with different speeds. For the Couzin swarm, after splitting, the agents moved at an average speed of 0.1m/s. The speed was reduced as the agents were crashing often at higher speeds. The time taken by Couzin model is higher than the sheep model, which was expected.

The videos of the shepherding model are presented in [34] for holonomic robots, [35] for non-holonomic Husky robot and the codes for the simulation in [36].

VI. CONCLUSION

In this paper, we analyzed the effect of using influential agents predators, leaders and a combination of predator and leaders to perform multiple tasks on three different swarm models namely, shepherding model, Couzin's model and physicomimetics models. The tasks were performed by splitting the swarm into sub-swarms using predators. The results show that the shepherding model with predator based splitting and steering performs the best compared to any other combination of leaders and predators for the other two swarm models. The physicomimetics takes the highest time to accomplish the mission. Although the shepherding model is the best, the model uses the center of mass to steer the agents under the influence of the predator. Hence, implementing such a system on robots is challenging due to visual constraints. Therefore, as a future work there is a further need to study the modification of this model for a real-world robotic application. Along with this, behaviour mixing of leaders and predators to coordinate and perform tasks will be studied.

REFERENCES

[1] A. Dirafzoon and E. Lobaton, "Topological mapping of unknown environments using an unlocalized robotic swarm," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 5545–5551.

[2] D. A. Shell and M. J. Mataric, "On foraging strategies for large-scale multi-robot systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 2717–2723.

[3] R. Groß, M. Bonani, F. Mondada, and M. Dorigo, "Autonomous self-assembly in swarm-bots," *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1115–1130, 2006.

[4] D. S. Calovi, U. Lopez, S. Ngo, C. Sire, H. Chaté, and G. Theraulaz, "Swarming, schooling, milling: phase diagram of a data-driven fish school model," *New journal of Physics*, vol. 16, no. 1, p. 015026, 2014.

[5] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *ACM SIGGRAPH computer graphics*, vol. 21(4). ACM, 1987, pp. 25–34.

[6] M. Nagy, Z. Ákos, D. Biro, and T. Vicsek, "Hierarchical group dynamics in pigeon flocks," *Nature*, vol. 464, no. 7290, pp. 890–893, 2010.

[7] R. Taylor, "A simulation model of locust migratory behaviour," *The Journal of Animal Ecology*, pp. 577–602, 1979.

[8] I. D. Couzin, J. Krause, R. James, G. D. Ruxton, and N. R. Franks, "Collective memory and spatial sorting in animal groups," *Journal of theoretical biology*, vol. 218, no. 1, pp. 1–11, 2002.

[9] A. Kolling, P. Walker, N. Chakraborty, K. Sycara, and M. Lewis, "Human interaction with robot swarms: A survey," *IEEE Transactions on Human-Machine Systems*, 2015.

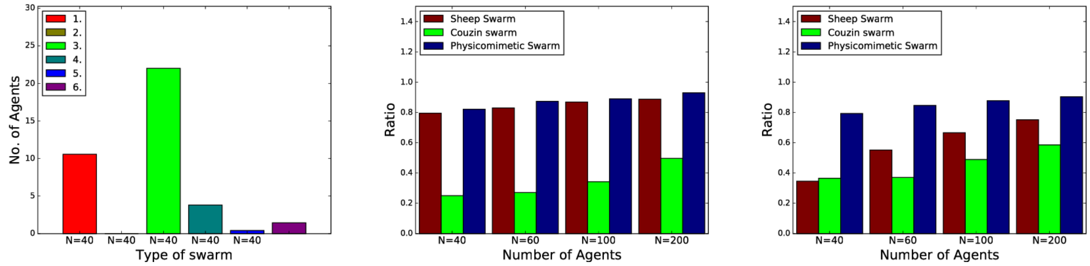
[10] M. A. Goodrich, P. B. Sujit, S. Kerman, B. Pendleton, and J. Pinto, "Enabling human interaction with bio-inspired robot teams: Topologies, leaders, predators, and stakeholders," BYU-HCMI Technical Report, Tech. Rep., 2011.

[11] I. D. Couzin, J. Krause, N. R. Franks, and S. A. Levin, "Effective leadership and decision-making in animal groups on the move," *Nature*, vol. 433, no. 7025, p. 513, 2005.

[12] S.-Y. Jung, D. S. Brown, and M. A. Goodrich, "Shaping couzin-like torus swarms through coordinated mediation," in *IEEE International Conference on Systems, Man, and Cybernetics*, 2013, pp. 1834–1839.

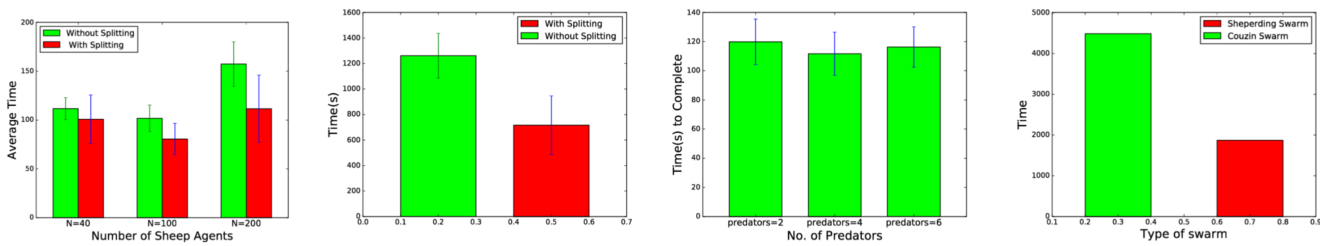
[13] X. Han, L. F. Rossi, and C.-C. Shen, "Autonomous navigation of wireless robot swarms with covert leaders," in *International Conference on Robot Communication and Coordination*. IEEE Press, 2007, p. 27.

[14] P. Walker, S. A. Amraii, M. Lewis, N. Chakraborty, and K. Sycara, "Control of swarms with multiple leader agents," in *IEEE International Conference on Systems, Man and Cybernetics*, 2014, pp. 3567–3572.



(a) (b) (c)

Fig. 8: (a) Average agents lost while completing the tasks for different configuration of leader and predators for each type of swarm. (b) The ratio of the number of agents in each group after splitting for different swarm models when speed of predators=0.1m/s (c) Effect of predator speed on splitting the swarm



(a) (b) (c) (d)

Fig. 9: (a) Average time taken by sheep swarm with four predators to complete all the tasks with varying number of agents. (b) Average time taken by sheep swarm to complete missions with more number of agents. Four predators have been used for 500 number of agents. (c) Effect of increasing the number of predators without splitting the swarm on a swarm of 40 agents. (d) Time taken by Couzin and Sheep swarm with 20 agents in Gazebo with husky (AGVs)

[15] K. Genter, S. Zhang, and P. Stone, "Determining placements of influencing agents in a flock," in *International Conference on Autonomous Agents and Multiagent Systems*, 2015, pp. 247–255.

[16] R. Tiwari, P. Jain, S. Butail, S. P. Baliyarasimhuni, and M. A. Goodrich, "Effect of leader placement on robotic swarm control," in *International Conference on Autonomous Agents and MultiAgent Systems*, 2017, pp. 1387–1394.

[17] K. Genter and P. Stone, "Influencing a flock via ad hoc teamwork," in *International Conference on Swarm Intelligence*. Springer, 2014, pp. 110–121.

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

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

[20] R. Vaughan, N. Sumpter, J. Henderson, A. Frost, and S. Cameron, "Experiments in automatic flock control," *Robotics and Autonomous Systems*, vol. 31, no. 1, pp. 109 – 117, 2000.

[21] T. Miki and T. Nakamura, "An effective rule based shepherding algorithm by using reactive forces between individuals," *International Journal of Innovative Computing, Information and Control*, vol. 3, no. 4, pp. 813–823, 2007.

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

[23] A. Özdemir, M. Gauci, and R. Gross, "Shepherding with robots that do not compute," in *Artificial Life Conference Proceedings 14*. MIT Press, 2017, pp. 332–339.

[24] C. K. Go, B. Lao, J. Yoshimoto, and K. Ikeda, "A reinforcement learning approach to the shepherding task using sarsa," in *IEEE International Joint Conference on Neural Networks*, 2016, pp. 3833–3836.

[25] B. Pendleton and M. Goodrich, "Scalable human interaction with robotic swarms," in *AIAA Infotech@ Aerospace Conference*, 2013, p. 4731.

[26] A. Pierson and M. Schwager, "Controlling noncooperative herds with robotic herders," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 517–525, April 2018.

[27] M. Chandarana, W. Luo, M. Lewis, K. Sycara, and S. Scherer, "Decentralized method for sub-swarm deployment and rejoining," in *IEEE International Conference on Systems, Man, and Cybernetics*, 2018, pp. 1209–1214.

[28] L. Chaimowicz and V. Kumar, "Aerial shepherds: Coordination among uavs and swarms of robots," in *Distributed Autonomous Robotic Systems 6*. Springer, 2007, pp. 243–252.

[29] I. Navarro, "Exploring the split and join capabilities of a robotic collective movement framework," *International Journal of Advanced Robotic Systems*, vol. 10, no. 5, p. 224, 2013.

[30] K. Raghuwaiya, J. Vanualailai, and B. Sharma, "Formation splitting and merging," in *International Conference in Swarm Intelligence*. Springer, 2016, pp. 461–469.

[31] Z. Chen, T. Chu, and J. Zhang, "Swarm splitting and multiple targets seeking in multi-agent dynamic systems," in *IEEE Conference on Decision and Control*, 2010, pp. 4577–4582.

[32] W. M. Spears, D. F. Spears, R. Heil, W. Kerr, and S. Hettiarachchi, "An overview of physicomimetics," in *International Workshop on Swarm Robotics*. Springer, 2004, pp. 84–97.

[33] https://youtu.be/xILUZXtkDlQ.

[34] https://www.youtube.com/watch?v=bY3Rbhq5nyA&t=55s.

[35] https://youtu.be/kYuvS6hSkOo.

[36] https://github.com/RaghavvGoel/HuskyAndMarkers.git.