The Shepherding Problem: A Comparison of Foundational Literature and Matlab Code

Captain Jake York (z5109674)
ZEIT3902 Engineering Research 3B
Australian Defence Force Academy

***Abstract*—In this paper, the model of the shepherding problem presented by Strömbom et al. [1] is compared with Matlab code provided by Prof. Abbass for accuracy and relevant differences. Although the solution to the problem was modelled in Matlab by the original authors, the code was not published alongside the paper. Therefore, the problem setup and the discussion of the results is used as the basis for the comparison to Prof. Abbass's code. The discussion and explanation in the original paper is fair, so comparison will be targeted at key areas of the problem and relevant differences, rather than any areas of misunderstanding or poor explanation.**

I. Introduction

The shepherding problem is a centuries-old concept based on the interactions between a herding sheepdog and a herded flock of sheep. In their paper, Strömbom et al. [1] describe the modelling of the problem as a key to understanding the behaviours of flocking animals and how robots can be used to influence these behaviours, as well as those of other artificial agents.

II. Problem setup

In the shepherding problem, there are two types of entity, the shepherd (which is usually a sheepdog in practice) and the flocking agents (usually sheep and in the paper, referred to as 'agents'). Hereafter, these will be referred to by the names shepherd and sheep.

In the model, a two dimensional square area of a pre-determined size (150 m) is used as the arena in which the agents interact. At the beginning of simulation, all of the sheep are spawned in the upper-right quadrant, while the shepherd is spawned in the lower-left. It is the goal of the shepherd to move all of the sheep to a pre-determined distance from the origin. Unlike it would be in reality, the arena is unbounded and both sheep and shepherd can move outside the bounds of the square area, although this is uncommon.

A. Sheep behaviours

The sheep in the model have several competing motives and behaviours that can be triggered. For these behaviours, relevant variables are distance, force and direction. These are technically in dimensionless units, although metres and timesteps are referred to in the paper. The behaviours of the sheep are quantified below:

- repulsion from other sheep: if any sheep moves inside a specified radius (r_a) of another sheep (2 m in the paper by Strömbom et al. [1]), then the active sheep is repelled with a force of 2 units in the opposite direction;
- repulsion from the shepherd: if the shepherd moves within a specified radius (r_s) of the sheep (65 m in the paper), then the active sheep is repelled with a force of 1 unit in the opposite direction;
- attraction to local centre of mass (LCM): if the shepherd is inside r_s of the active sheep, then the sheep is attracted to the LCM of its n closest neighbours with a force of 1.05 units;
- attraction to previous direction of movement: in order to smooth out the turning of the sheep, they have an attraction to their previous timestep's direction of movement of 0.5 units;
- noise: to simulate reality, each sheep has angular noise of 0.3 units configured to prevent deadlock scenarios and simulate imperfect agents.

The sheep move a distance of 1 m per timestep. They also have a 5% probability of moving without interaction by the shepherd to simulate realistic grazing behaviour. The simulated numbers of sheep were between 20 and 201.

B. Shepherd behaviours

Much like the sheep, the shepherd experiences competing motives and behaviours. These are quantified below:

- herding positioning: if the collective body of sheep is aggregated within a given radius ($r_a N^{2/3}$) of the flock global centre of mass (GCM), then the shepherd moves to a point that is a distance $r_a sqrt(N)$ from the flock and places the flock directly between the target location and the shepherd;
- collecting positioning: if any sheep are not correctly aggregated, then the shepherd moves to a point that is r_a distance from the furthest sheep from the GCM and places the sheep directly between the GCM and the shepherd;
- herding: if the sheep are aggregated and the shepherd is correctly positioned, the shepherd then moves towards the target location, forcing the herd to also move in that direction;

Submitted 7 August 2018

- collecting: once the shepherd has completed collecting positioning, the shepherd then drives the relevant sheep towards the GCM to aggregate them (it is then common to see the shepherd immediately move to a driving position and drive the herd);
- movement prevention: if the shepherd is within $3r_a$ of any sheep, it is stationary for the current timestep;
- noise: as per the sheep, the shepherd experiences noise in all movements.

The shepherd moves a distance of 2 m per timestep. The shepherd has a blind spot past $\pi/2$ radians from its direction of view. This was introduced to prevent the shepherd being caught between smaller groups of sheep and not being able to resolve the situation. The authors experimented with the shepherd being tasked with bringing in smaller groups of sheep at a given time, with the greatest success rate for $n = 20$. There is only ever one shepherd simulated by Strömbom et al. [1]

III. Key Similarities

There are several key similarities between the original model and the Matlab code presented by Prof. Abbass. His work follows the guidelines presented in the model by the original authors. The similarities are presented in this section below:

- key problem parameters: the problem setup, shepherd and sheep have retained all of their key parameters from the original model, such as interaction distances and prescribed numbers of sheep;
- terrain: as per the original model, Prof. Abbass's Matlab code is presented as a two dimensional borderless problem that does not consider height of terrain and potential areas to slow down/speed up movement.

IV. Key Differences

There are key differences between the two models, the details of which are presented here:

- sheep direction vectors: Prof. Abbass's implementation develops on the original by showing the directions of influence on each sheep during each time step;
- shepherd numbers: unlike the original paper, Prof. Abbass's code allows for multiple shepherds, which are repelled from each other when inside a designated minimum distance. Although this feature has been added, an anomaly was found where a second shepherd does not effectively drive the herd of sheep with the first and reduces effectiveness after initial collection of the herd (see Figure 1);
- behaviour of sheep: although the code obeys the principles presented in the original paper, the sheep in Prof. Abbass's work exhibit a stronger centre of gravity that repels the influence of the shepherd. With large total sheep numbers, the shepherd is actually forced away from the target as if the sheep are intent on not being herded. This is shown in the comparison in Figures 2 and 3 where the shepherd repulsion radius is reduced to only 10 m;
- blind angle and herding packets: although they are mentioned as not implemented in Prof. Abbass's code, it

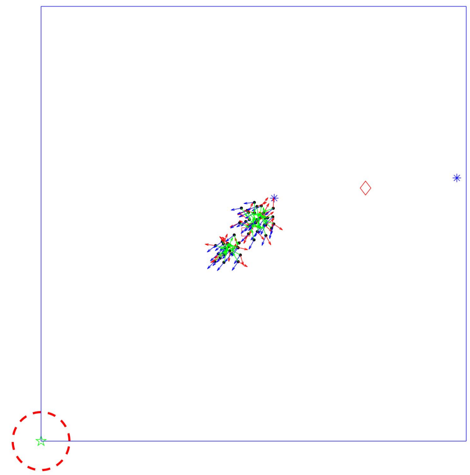

Fig. 1. Second shepherd providing reduced benefit

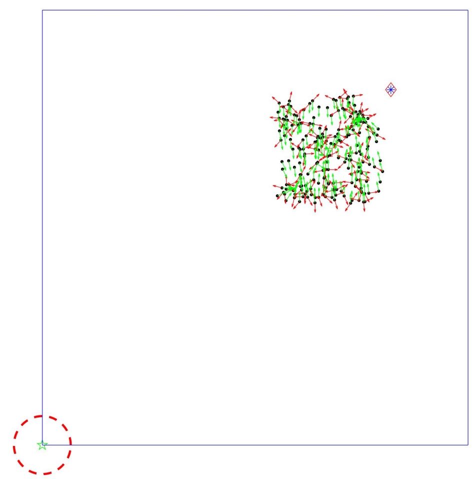

Fig. 2. Shepherd repulsion radius reduced to 10 m - sheep in approximately starting locations (ticks < 30)

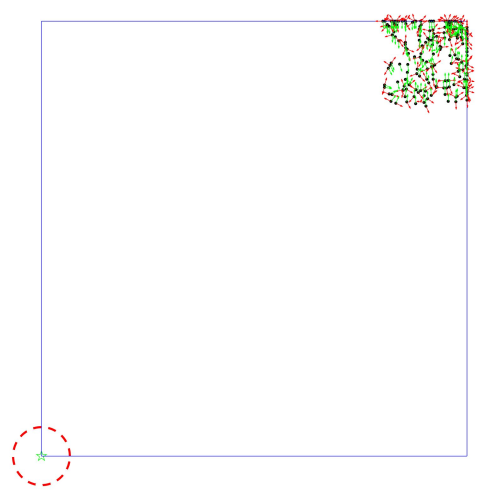

Fig. 3. Shepherd repulsion radius reduced to 10 m - sheep force shepherd away from target position (ticks > 100)

bears mentioning here that these are a key difference. The shepherd in Prof. Abbass's implementation of the model has no blind spot behind it and will only ever operate on the entirety of the herd - there is not packet breakdown that occurs to reduce the amount of sheep to herd at a single point in time;

- Prof. Abbass's shepherds do not approach the sheep herd directly - they will move directly along the x or y directions to bypass the herd and move towards the driving position. It is through this behavioural trait that the implementations can be shown to be different. Although this difference means that the shepherds will not inadvertently push the herds away from the target position, it will overall cost time. Prof. Abbass's shepherds will not collect the sheep as a first task if they are not aggregated enough - they will move to the driving position and then do smaller collection movements from that position. This becomes ineffective when the sheep herd is dispersed and the paddock size is large (test case was 1500 m).

V. Discussion

Prof. Abbass's code implementation functions in some ways like the model designed and presented by Strömbom et al. [1]. However, the implementation has several differences that either reduce or enhance the power of their implementation.

By increasing the number of shepherds available, different dynamics can be explored between the two types of agent. However, in this respect, the implementation can be refined to ensure that the shepherds work together in the most efficient way. This could be done by distributing collection and driving tasks, ensuring all-round coverage of the shepherded herd to maintain collection or by driving in marginally opposing directions to double the driving action. If the latter was the case, it would be realistic to allow the sheep to move at a greater rate than 1 m per time tick.

As Prof. Abbass's shepherds do not approach the sheep herd directly, they will not cause aggregation of the sheep as a byproduct of moving to the driving position. They will then not carry out effective collection if the herd is not yet aggregated. This requires further analysis to implement a fix.

The paddock size in the video simulations provided by Strömbom et al. [1] is the same as the test cases used by Prof. Abbass. Although different paddock sizes are not shown in the supplementary materials of the original model, increasing the size can be used to break the implementation of Prof. Abbass's model.

The sheep in Prof. Abbass's implementation, when within a certain paddock size (causing a starting level of aggregation), will have a tendency to drift away from the target position. When the shepherd's influence radius, r_s, is reduced (to, for example, 10 m), then the sheep can eventually force the shepherd away from the target position because, as mentioned above, the shepherd will move around the herd to be opposite the target position before collecting or driving. The shepherd's force on the collective body of sheep is not enough to prevent their drifting away from the target position and it can eventually be forced past the boundary of the paddock.

In both the original model and in Prof. Abbass's code, development and learning is not implemented. The agents follow strict behaviours and are not able to learn from previous actions in order to improve performance. If this was possible, improvements to the model and insights about changes to the behaviours could be found.

VI. Conclusion

The implementation by Prof. Abbass is different to the model shown and described by Strömbom et al. [1], but using initial test cases, exhibits largely the same outcomes. After testing the behaviours by changing some of the variable inputs, they can be broken and the shepherd can be overwhelmed by the task set before it. After further analysis and testing, it may be possible to rectify these issues so that the shepherd performs better in all realistic cases.

References

[1] Daniel Strömbom, Richard Mann, Alan Wilson, Stephen Hailes, Jennifer Morton, David Sumpter and Andrew King. *Solving the shepherding problem: heuristics for herding autonomous, interacting agents*. Journal of the Royal Society Interface, Volume 11, Issue 100, 2014.