Robot Control of Animal Flocks

Richard Vaughan[1,2], Neil Sumpter[1], Jane Henderson[1], Andy Frost[1] & Stephen Cameron[2]

[1]BioEngineering Division, Silsoe Research Institute, Silsoe, Bedford MK45 4HS, UK.
[2]Oxford University Computing Laboratory, Parks Road, Oxford OX1 3QD, UK.

email: Richard.Vaughan@comlab.ox.ac.uk

ABSTRACT

The Robot Sheepdog Project has developed a mobile robot that gathers a flock of ducks and manoeuvres them safely to a specified goal position. This is the first example of a robot system that exploits and controls an animal's behaviour to achieve a useful task. A potential-field model of flocking behaviour was constructed and used to aid the design of two novel flock-control methods. These methods are described and evaluated in a series of simulated and real-world experiments.

1 INTRODUCTION

This paper presents a complete robot system that controls the behaviour of another intelligent system in the presence of variability, uncertainty and noise. Our Robot Sheepdog Project has demonstrated a robot that gathers a flock of ducks and manoeuvres them safely to a pre-determined goal position.

The sheepdog's gather-and-fetch task was chosen because of its familiarity and the strong interaction between the dog, shepherd and flock animals. Using ducks instead of sheep allows us to experiment on a conveniently small scale, in a controlled indoor environment. Duck flocking behaviour is recognised by shepherds as similar to sheep; ducks are often used to train sheepdogs because of their relatively slow movement.

In order to identify the appropriate robot-animal interactions we built a minimal generalised model of the underlying flock behaviour. The hypothesis is that if the model accurately captures the basis of the behaviour, then a system which controls the model should control the real-world behaviour.

Models of flocking behaviour exist in the literature and are generally derived from Hamilton's observation that flocking may be produced by the mass action of individual animals, each seeking the proximity of its nearest neighbours [2]. It was later suggested that this behaviour can be well modelled by an attractive 'force' acting between the animals, with the magnitude of the attraction varying with the inverse square of the animals' mutual distance [3] [5]. It is argued that this relationship represents a linear response to sensory information which itself varies with the inverse square of distance. Similar models have produced realistic computer animations of bird flocks [4].

These ideas are familiar in robotics, where such *potential field* techniques are used for navigation [1, Ch.10-11]. This class of algorithm uses the analogy of forces acting on particles, such that the robot will move as if it were a particle attracted or repelled from features in its environment. A robot is typically attracted to a goal position and repelled from obstacles.

The commonality of these animal and robot behaviour models forms the basis of two effective flock-gathering strategies, described below.

2 A ROBOT SHEEPDOG

The experimental system comprises a robot vehicle, a workstation and a video camera (Figure 1, left). The vehicle was designed to work in a duck's environment: outdoors, on short grass, and in real time. Thus our robot has an acceleration $\approx 1ms^{-2}$ and a top speed $\approx 4ms^{-1}$, which is about twice as fast as the ducks. It is covered in a soft plastic bumper mounted on rubber springs, ensuring duck safety. In the tradition of mobile robotics, we call it 'Rover' (Figure 1, right).

The vehicle and ducks are free to move in a visually uniform arena of 7m diameter, in view of the overhead camera. The position and orientation of the robot, and the position and size of the flock are determined by processing the video image stream. Standard background-differencing and region-growing techniques are used to achieve an update rate of approximately 20Hz. Note that the vision system determines only the center of mass (and radius) of the flock and not the positions of individual birds. It was found that the vector from robot to flock center closely approximates

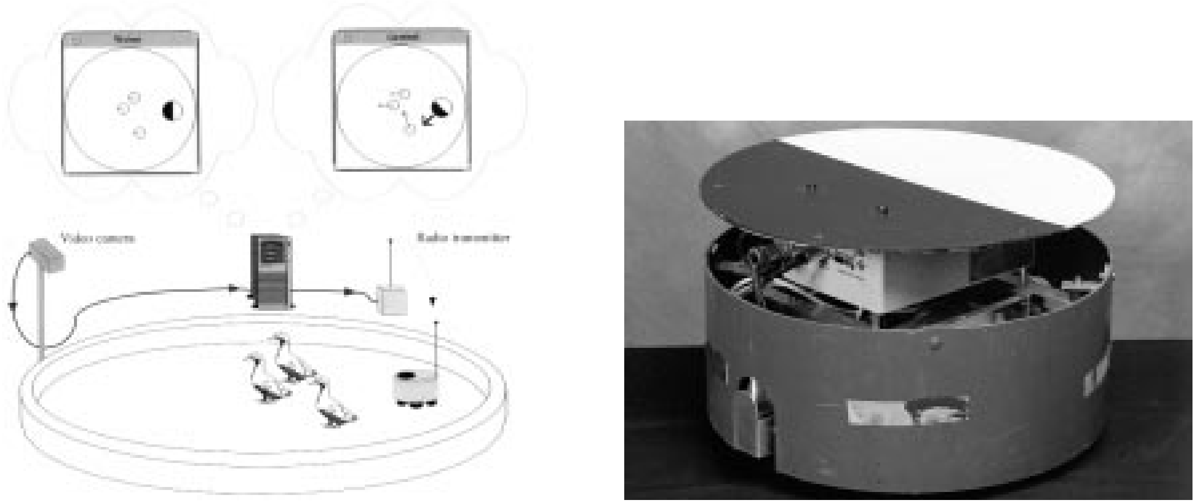


Figure 1: Robot Sheepdog system overview (left) and vehicle (right)

the resultant of the vectors from the robot to each bird, and this information has proved to be sufficient for these experiments.

The robot's high-level controller (running on the workstation) instantiates a 'flock-control algorithm' which steers the robot to gather the flock and return it to a goal position on the edge of the arena. This algorithm takes the vision data (positions of the robot R, flock F and goal G) as input and returns a desired vehicle trajectory $(R, F, G) \rightarrow \vec{r}$.

Wall collisions are prevented by modifying any $\vec{r}$ that would take the robot outside the arena, by shifting its end point inside the boundary along the arena radius.

The vehicle's current observed speed and heading are compared to $\vec{r}$, and new goal wheel-speeds are determined by the function

$$\begin{pmatrix} R_{left} \\ R_{right} \end{pmatrix} = \begin{pmatrix} |\vec{r}| + \frac{D}{2}(\theta - \angle\vec{r}) \\ |\vec{r}| - \frac{D}{2}(\theta - \angle\vec{r}) \end{pmatrix}$$

where θ is the robot's current heading, D is the distance between the wheels and .$\angle\vec{r}$ is the direction of These wheel-speed demands are passed to the vehicle via a radio modem.

The vehicle's hardware runs the proportional controller

$$u(t) = K(R(t) - E(t))$$

(where u is the output to the wheel, R is the desired speed, E is the speed error, and K is the controller gain) for each wheel at 100Hz. The controller gain was chosen by experiment and it is found that the vehicle's movement closely approximates the desired movement vector $\vec{r}$.

3 A MODEL FLOCK

A minimal simulation model of the duck-herding scenario was created, in which a flock of model ducks (ducklets) moves in a circular arena containing a model robot.

Given a ducklet's position D, the positions of the N other ducklets $D_{1 \rightarrow N}$, the robot's position R and the nearest point on the wall W, the ducklet's movement vector $\vec{d}$ is determined by the function shown in Figure 2. The ducklets are (1) attracted to each other, aggregating the flock; (2) repelled from each other, preventing collisions and maintaining inter-ducklet spacing; (3) repelled from the arena wall, preventing collisions. A further term (4) which produces repulsion from the robot is proposed to model the aversive response of the ducklets to the robot. All these forces are scaled according to the inverse square of distance, and each ducklet moves according to the resultant of the forces acting upon it. The simulation produces a realistic-looking flock which can be manipulated by steering the model robot.

Note that the model describes a small subset of the ducks' behaviour. Of course, many other mechanisms generate the behaviour of real ducks, but our hypothesis is that this model captures enough of the real animals' behaviour to be a useful design tool. The model is a *generalised* description of flocking behaviour and as such could be applied to any flocking animal in two or three dimensions.

4 PROCEDURE

Experiments with the simulator guided the development of a novel flock control algorithm which is closely related to the flock model described above. A series of experiments were performed to assess the flock-control method.

4.1 Simulation

The algorithm is first tested in simulation. A point on the arena boundary is chosen as the flock goal, 12 ducklets are placed randomly in the arena, and the robot positioned near the goal. The simulation starts and the positions of the robot and flock center are recorded for the next 3 minutes, as the robot attempts to manoeuvre the flock to the goal.

The experiment was repeated nine times with the ducklets at different random start positions, and the robot at a slightly different position near the flock goal in each trial.

4.2 Real world

A similar experiment was then performed in the real world. A random point along the arena boundary is chosen as the flock goal. With the robot inactive and positioned near the goal, a flock of 12 ducks is introduced into the arena. After 3 minutes accomodation time, the robot is activated. The positions of the robot and flock center are recorded for the next 3 minutes, as the robot attempts to manoeuvre the flock to the goal. At the end of the trial, the robot is deactivated and the ducks move freely again for 2 minutes before being allowed out of the arena.

The experiment was repeated three times with each of three flocks, with the robot at a slightly different position near the flock goal in each trial. Multiple flocks were used to increase the chance of variability in behaviour between trials. All the ducks were the same age and had been raised under similar conditions.

5 EXPERIMENT 1

5.1 Algorithm

The robot's movement vector $\vec{r}$ is given by the function shown in Figure 3. The robot is (1) attracted to each ducklet with a magnitude proportional to their mutual distance. This force causes the robot to move towards the flock. A second force (2) repels the robot from each ducklet with a magnitude proportional to the inverse square of their mutual distance. This prevents collisions. The resultant of these two forces creates a circular orbit of zero potential around the flock center. A further force (3) repels the robot from the goal position with a constant magnitude. This has the effect of tilting the potential landscape such that the orbit around the flock now has a minimum behind the flock with respect to the goal. The robot will move towards this point; driving the flock away from it and towards the goal.

5.2 Simulation results

These results show that the controller performs the required task with some success. Figure 5 shows a representative plot of the simulated robot and flock paths around the arena (A). A characteristic robot behaviour emerges in which the robot moves away from the goal towards the ducklets, initially pushing them away from the goal until they meet the wall of the arena. The robot then moves around behind the flock with respect to the goal. The flock moves away from the robot, hence towards the goal, and pulls the robot with it. As the flock approaches the goal, the goal repulsion acting on the robot causes the robot to 'stand-off', increasing the distance between the flock and the robot and decreasing the 'push' on the ducklets. The ducklets overshoot the goal slightly and the robot again moves towards them to push them back. This produces an oscillatory motion, with the flock moving back and forth across the goal, and the robot moving in a figure-of-eight pattern to keep them there. The size of the oscillations decreases over time, so the system is stable about the goal position.

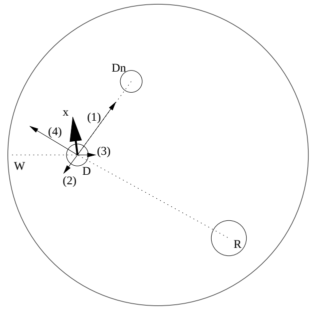
The plot below the arena map in Figure 5 (B) shows the distance of the flock to the goal over the length of the trial, plus the average distance over the entire trial. This is used as a measure of the trial's success. This trial scores an average flock-to-goal distance of 2.2m. The average score over 9 trials was 2.2m.

5.3 Real world results

Figure 5 shows a representative plot of the real robot and flock paths around the arena (C). It can be seen that the flock is brought near the goal, but overshoots and must be fetched back by the robot, again it overshoots and the system oscillates with the ducks around the goal point and the robot tracing a similar characteristic path to the simulation. The success plot (D) clearly shows the oscillatory behaviour. This trial scores an average flock-to-goal distance of 1.46m. The average score over 9 trials was 2.65m.

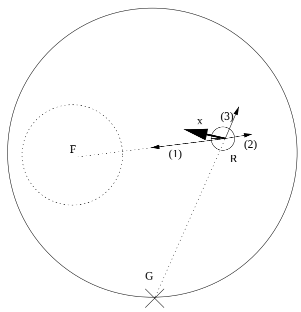
5.4 Discussion

Further trials with the simulator have shown that the size of this oscillation can be reduced, and thus performance enhanced, by tuning the robot-to-flock attraction gain (K_1 in Figure 3). This effectively controls the distance that the robot keeps from the flock, which is related to the 'push' exterted on the ducks to control their movement. The success of the system is sensitive to changes in this parameter. However the large variation we find between flocks and over time means that to achieve good success, and in particular to make the ducks settle near the goal, would mean tuning the parameter for each trial. This could perhaps be



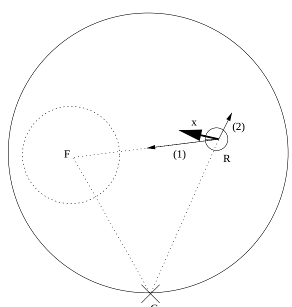
$$\vec{d} = \sum_{n=1}^{N}\left(\underbrace{\left(\frac{K_1}{|\vec{DD}_n|^2}\right)\widehat{DD_n}}_{(1)} - \underbrace{L - \left(\frac{K_2}{|\vec{DD}_n|^2}\right)\widehat{DD_n}}_{(2)}\right) - \underbrace{\left(\frac{K_3}{|\vec{DW}|^2}\right)\widehat{DW}}_{(3)} - \underbrace{\left(\frac{K_4}{|\vec{DR}|^2}\right)\widehat{DR}}_{(4)}$$

Figure 2: Flock model (schematic not drawn to scale). Key: gain parameters $K_{1\to 4}$; shift parameter L; ducklet position D, other ducklet D_n; Robot position R; Nearest point on wall W; algorithm terms $(1 \to 4)$ and resultant $\vec{d}$ (where $\widehat{a}$ is the unit vector of $\vec{a}$).



$$\vec{r} = \underbrace{K_1\vec{RF}}_{(1)} - \underbrace{\left(\frac{K_2}{|\vec{RF}|^2}\right)\widehat{RF}}_{(2)} - \underbrace{K_3\widehat{RG}}_{(3)}$$

Figure 3: Method 1 (schematic not drawn to scale). Key: gain parameters $K_{1\to 3}$; flock center F; Robot position R; Goal position G; algorithm terms $(1 \to 3)$ and resultant $\vec{r}$ (where $\widehat{a}$ is the unit vector of $\vec{a}$)



$$\vec{r} = \underbrace{(K_1|\vec{GF}|)\widehat{RF}}_{(1)} - \underbrace{K_2\widehat{RG}}_{(2)}$$

Figure 4: Method 2 (schematic not drawn to scale). Key: gain parameters $K_{1,2}$; flock center F; Robot position R; Goal position G; algorithm terms $(1 \to 3)$ and resultant $\vec{r}$ (where $\widehat{a}$ is the unit vector of $\vec{a}$)

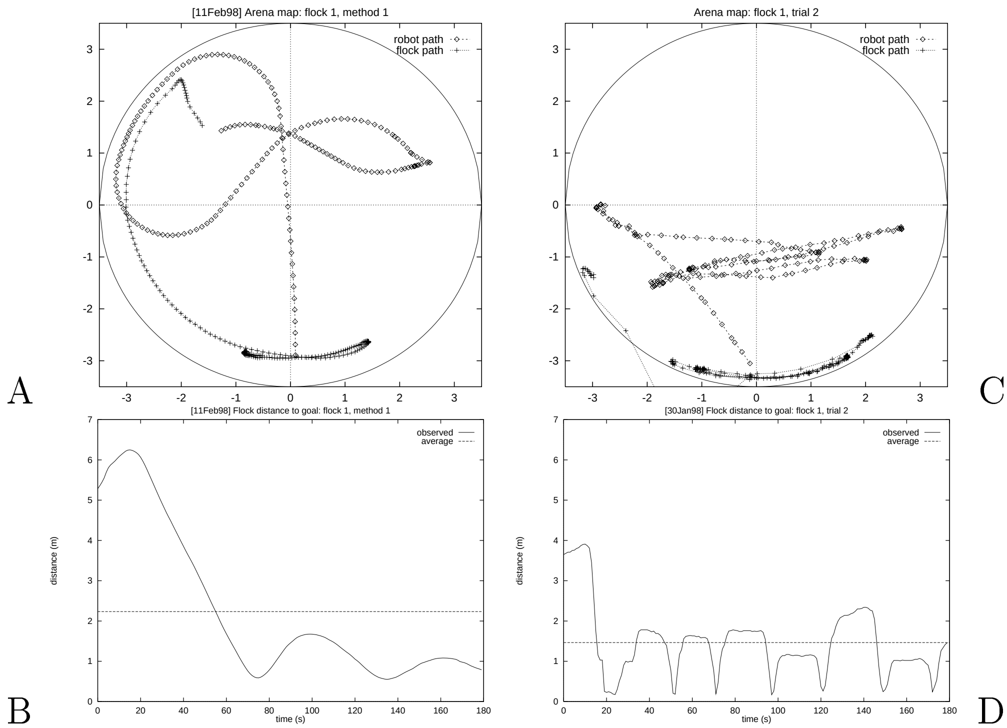


Figure 5: Method 1 results for simulation (left) and real world (right)

achieved with an appropriate adaptive algorithm, but consideration of these results suggested another, simpler flock control algorithm which avoids this problem.

6 EXPERIMENT 2

The distance $|GF|$ (see Figures 3 & 4) is the system variable we are trying to control, ie. reduce to zero. In a classical proportional controller a control output would be applied to correct this variable, with a magnitude proportional to the size of the error. If we introduce this term into the flock controller, we can design an analogous system whereby the repelling stimulus experienced by the ducks is proportional to their distance from the goal. This should reduce the problem of oscillation about the goal caused by an excessive control signal.

6.1 Algorithm

The robot's movement vector $\vec{r}$ is given by the function shown in Figure 4. The robot is (1) attracted to the flock with magnitude proportional to the distance from the *flock to the goal*; (2) repelled from the goal with constant magnitude.

6.2 Simulation results

Figure 6 shows a representative plot of the simulated robot and flock paths around the arena (A). It can be seen that the flock is brought near the goal. The success plot (B) shows that the oscillatory behaviour of Method 1 does not occur. This trial scores an average flock-to-goal distance of 1.9m. The average score over 9 trials was 1.8m.

6.3 Real world results

Figure 6 shows a representative plot of the real robot and flock paths around the arena (C). It can be seen that the flock is brought near the goal, but overshoots and must be fetched back by the robot. However the flock eventually settles near the goal and the robot retreats to the opposite side of the arena. The success plot (D) clearly shows the initial overshoot, followed by the stable behaviour. This trial scores an average flock-to-goal distance of 1.59m. The average score over 9 trials was 2.27m.

Subsequent trials (simulated and real) have shown that the size of the overshoot can be greatly reduced by tuning the gain parameter K_1. As for Method 1 the optimal setting varies for different flocks, but this has a less significant effect on the overall success, and the general performance is therefore increased.

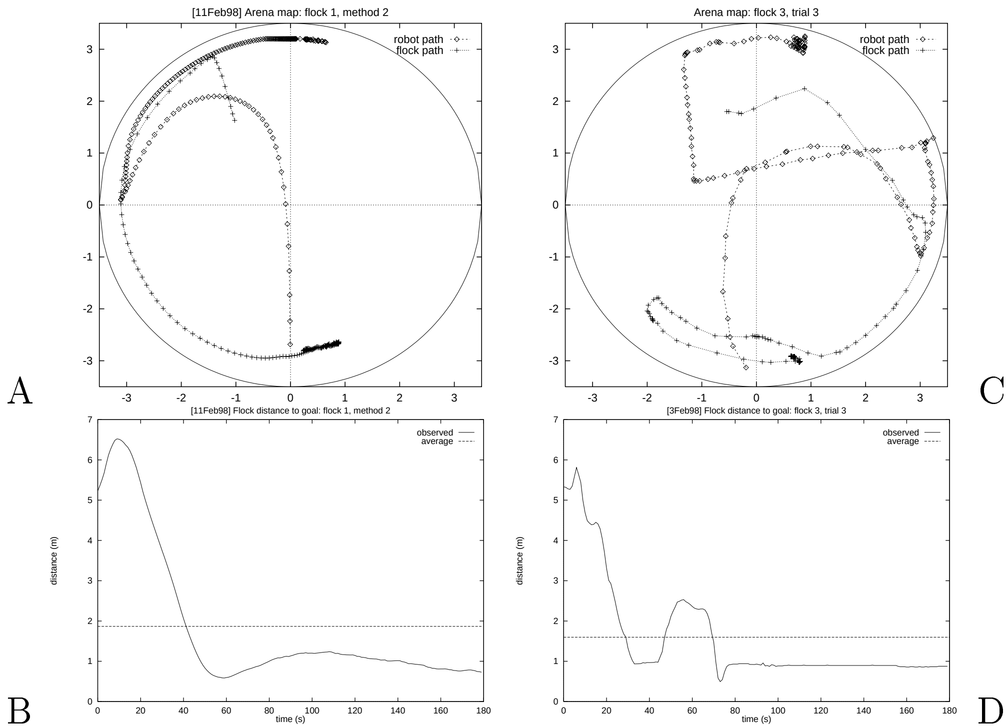


Figure 6: Method 2 results for simulation (left) and real world (right)

7 CONCLUSION

We have demonstrated a robot system that achieves a sheepdog-like task, gathering and fetching live animals to a pre-defined goal position. We believe this is the first automatic system to exploit an animal's behaviour to achieve a useful task. A flock control method was designed and tested using a minimal simulation model of the ducks' flocking behaviour, and successfully transferred to the real world. These experiments led to the design of a second, simpler method with improved performance. We assert that the effectiveness of the simple methods described is due to their close relationship to the mechanisms underlying flocking behaviour itself, and conclude (1) that behavioural simulations can be plausible engineering design tools, and (2) that such a methodology is appropriate for future animal-interactive robotics experiments.

ACKNOWLEDGEMENTS

Thanks to Steve Crook, Paul Griffith, John Lowe and Robert Smith for their work on the robot, and to two anonymous reviewers for their helpful suggestions on this paper. The Robot Sheepdog Project is funded by the BBSRC and Silsoe Research Institute.

REFERENCES

[1] Stephen Cameron and Penelope Probert. *Advanced Guided Vehicles: Aspects of the Oxford AGV project.* World Scientific, 1994.

[2] WD. Hamilton. Geometry for the selfish herd. *Journal of Theoretical Biology*, 31:295–311, 1971.

[3] B. L. Partridge. The structure and function of fish schools. *Scientific American*, pages 114–123, June 1982.

[4] Craig W. Reynolds. Flocks, herds and schools: A distributed behavioural model. *Computer Graphics*, 21(4):25–34, jul 1987.

[5] K. Warburton and J. Lazarus. Tendency-distance models of social cohesion in animal groups. *Journal Of Theoretical Biology*, 150:473–488, 1991.