Experiments in Animal-Interactive Robotics
D.Phil. Thesis

Richard Vaughan
Oriel College, Oxford

Abstract

This thesis describes the development of an autonomous robot system that gathers a flock of ducks in a circular arena and manoeuvres them safely to a pre-determined goal position. In the process it establishes a methodology for developing robots that interact with animals. An important feature of this methodology is that it enables the development of a machine that can usefully interact with an animal *without using the animal in the design process.*

Interacting with animals imposes strong constraints of real-time action, robustness and animal safety. A suitable arena, robot vehicle, control architecture and vision system are described.

An animal-interactive robot must be robust with respect to the inevitable variations in behaviour between individual animals and even in the same animal over time. It is suggested that (a) animal-interactive robot controllers should exploit the underlying mechanisms of the subject animals' behaviour rather than the details of any particular animal or group, and therefore (b) a simple model of such an underlying mechanism can be used to aid the design a robot that will control the real animal system. Specifically, it was hypothesized that a robot controller that reliably gathers a simulated flock should also work when transferred to a real robot and flock of ducks.

A minimal generic flocking model is created and incorporated into a simulation of a the vehicle and arena. A simple robot controller is devised and demonstrated to work in simulation and transfer directly into the real world. A series of experiments are performed to assess the performance and reliability of the method. Consideration of these results leads to a second, simpler algorithm. The second method is tested in simulation and the real world and found to be more successful and more reliable than the first. A pair of no-robot control experiments gives benchmarks with which to compare the performance of the two methods.

Further experiments in simulation demonstrate the application of the flock control methods to on- robot sensing modalities; laser ranging and vision.

The thesis demonstrates the first robot interacting with animals to achieve a useful task. It is concluded that a minimal behavioural simulation can be a useful design tool for a real animal-interactive robot system.

Acknowledgements

The work was undertaken as part of the Robot Sheepdog Project; run by the BioEngineering Division of Silsoe Research Institute, Silsoe, Bedfordshire, UK. in association with the Universities of Oxford, Bristol and Leeds. A BBSRC studentship paid the bills.

Thanks to my colleagues Jane Henderson and Neil Sumpter, to Stephen Cameron, Andy Frost, Nick Tillett and Christopher Wathes for their supervision of me and the project, Len Burgess for his craftsmanship and to Steve Crook, Paul Griffith and John Lowe at SRI for building us a robot. Robert Smith provided valuable assistance in the dark debugging days.

I'm grateful to Gaurav Sukhatme and Maja Mataric for inviting me to work with them and allowing me time to finish this thesis. Having moved so far from home I'm already missing Jim Alford, Ben Southall, Ross Cant, Dave Abrutat, Pat & John and the many stupid late nights in Silsoe.

I'm in this business thanks to the friendship and inspiration of Seth Bullock, Mike Wheeler and Phil Husbands, but its thanks to Tricia (and Jo, Rose and Mike) that I'm in any business at all.

Last and most thanks goes to Julia for everything.

Dedicated to my grandfather, David, who likes to know how things work.

Richard Vaughan, Los Angeles, November 1998.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The Robot Sheepdog Project

Robots have already found a place in animal husbandry through automatic milking systems [Prescott, 1995] and robotic sheep shearing [Trevelyan, 1989]. There may be a role for robots in agricultural systems, replacing humans in hazardous, tedious or unpleasant tasks, or where there are benefits in efficiency, effectiveness and animal welfare. Duncan et.al. have shown that an appropriately designed machine can harvest broiler chickens more efficiently than humans *and* cause less stress to the birds [Duncan et al., 1987].

The task of designing robots to interact with animals raises issues not encountered in other industries. First, the welfare of animals must be considered; robots must be designed to cause minimum stress to the animals they work amongst. Secondly, unlike the objects of typical robot applications, animals are autonomous agents and will exhibit behaviour. This is both a problem and an opportunity. For example most animals can move themselves around: they may not stay where you left them, but on the other hand they may move to where you want them. Thus animals need not be handled directly, but can (and perhaps should) have their *behaviour* manipulated towards achieving some goal. We believe that investigating such *animal-interactive* systems is an interesting area for research in both robot design and animal behaviour, and this project was conceived to examine these issues.

1.1.1 Goal

The goal of this work is to demonstrate the first active control of an individual or group of live animals by interacting with their natural behaviour in a loosely-constrained environment.

During the proposal stage of this project informal pilot trials showed that it was possible for a human operator to manoeuvre a small flock of ducks close to a pre-determined goal point using a radio-controlled car. The task proved difficult for the operator and success varied between trials, but it was shown that the ducks could be controlled in a useful way by interaction with a small vehicle. This task was chosen as the exemplar which would demonstrate robot control of animals by behavioural interaction.

This task presents a major difference from conventional robot applications in that the objects cannot be manipulated directly, but must be influenced to move *themselves* to the goal. We sought to identify those aspects of duck behaviour which make them controllable, and to design a herding strategy to exploit those features and effectively control the flock.

1.1.2 Organisation

The Robot Sheepdog Project (hereafter RSP) was conceived and coordinated by researchers at the BioEngineering Division of Silsoe Research Institute, Silsoe, Bedfordshire, UK. (hereafter SRI). The project comprised three PhD/DPhil studentships:

> Richard Vaughan, *Animal Interactive Robotics*, supervised by Andy Frost, Nick Tillett (SRI) and Dr. Stephen Cameron (Oxford University Computing Laboratory).
>
> Neil Sumpter, *Recognition and Tracking of Individuals and Groups of Animals*, supervised by Dr. Robin Tillett (SRI) and Dr. Roger Boyle (School of Computer Studies, University of Leeds).
>
> Jane Henderson, *Adaptive Responses of Herding Animals to a Robot Vehicle*, supervised by Prof. Christopher Wathes, Dr. Jeff Lines (SRI) and Dr. Christine Nichol (Dept. of Clinical Veterinary Science, University of Bristol)

The team met every three months to assess progress and coordinate our efforts. Each student had a distinct subject area; we overlapped little but complemented each other to form a broad study. The central goal of the project was the demonstration of a 'Robot Sheepdog' which could exhibit some degree of control over the ducks. This thesis describes the development of that robot.

1.2 Thesis outline

Chapter 2: Animal-Interactive Robotics

The robot's task is described and this work motivated by possible benefits to welfare and efficiency in animal-handling systems.

Relevant literature is surveyed and it is concluded that the goal and execution of this project are unique. However, material from many fields provides a solid starting point for this research, including flock modeling work from biology and computer graphics, and both conventional and behavioural robotics research.

The approach to the problem is described and placed in the context of behavioural robotics. In general, animal-interactive robot systems must identify *reliable behaviours* in the subject animals, and produce *appropriate interactions* to exploit them.

Three hypotheses are presented: (1) that flock control can be achieved by exploiting the animals' threat-avoidance behaviour; (2) that the appropriate interaction is to place the robot behind the flock with respect to the goal, at some appropriate distance; (3) that such a behaviour could be designed in simulation and transferred directly to the real world.

Chapter 3: Rover the Robot Sheepdog

The design and development of the robot system is described, including a purpose-built vehicle, an experimental arena and an overhead vision system for tracking the positions of the flock and robot. To suit the application, the vehicle is deliberately simple, cheap and robust; the inherent uncertainty in the overall system precludes the use of high-precision robot engineering.

Effective control of the vehicle is demonstrated with a series of trials in which the robot traces shapes around the arena and recovers from small disturbances.

A generic flocking model is created as a test-bed for flock-control experiments.

As the main experiments were performed, the vision system suffered reliability problems from an unusual source. Appendix A reports the solution.

Chapters 4 & 5: Experiments

A simple potential-field based robot control strategy is proposed and found to successfully gather the simulated flock. Results of several trials are presented and metrics are devised to assess performance. The identical controller is run on the real robot and tested with real flocks of ducks. The results are less reliable than in the simulation, but do demonstrate the first autonomous robot control of animal behaviour.

Consideration of these results leads to an improved proportional control-style strategy which is found to be superior in simulation. Real-world experiments with this strategy prove to be far more successful and reliable than with the original.

Chapter 6: Discussion

The results from the previous chapters are compared and the reasons for the discrepancy between simulated and real world trials are discussed; the main causes are found to be a time-scale mismatch between the two, plus tracking errors and delays in the real system.

A pair of control trials are described; a null experiment in which the flock receives no stimulus, and a trial in which the flock is attracted to food. These establish benchmarks for success against which the results from the previous experiments are compared.

The limitations of the experiments and strategies are discussed and some questions and criticisms are addressed. Suggestions are made for extensions to the work such as dealing with corners and using multiple robots.

The robot uses a global overhead position sensor which would not be feasible in many situations. It is demonstrated in simulation that the robot controllers can be modified to use a mixture of local and global sensing (local rangefinder + global goal-direction), or purely local (on-board vision) sensing.

Chapter 7: Conclusion

Finally, the conclusion compares the stated goals to the results. It is suggested that the hypotheses stated in Chapter 2 were proven, and that this work does indeed demonstrate effective control of live animals by interaction with an autonomous robot.

This thesis provides:

1. a methodology for experiments in robot/animal interaction.
2. an appropriate robot and control architecture for experiments in autonomous vehicle/duck interaction.
3. strategies for controlling the movement of flocks using an autonomous robot vehicle.

Opportunities for extending the abilities of this robot are discussed, along with suggestions for future animal-interactive robot projects.

Chapter 2

Animal-Interactive Robotics

This chapter describes the goals of this work, including the unique task that the robot performs. The project is placed in the context of behaviour research from the biological, agent-centered AI and robot literature. The general approach is described and related to that of behavioural robotics. A set of hypotheses is generated, to be tested in the body of the thesis.

2.1 Statement of research problem

The shepherd/sheepdog team can display remarkable control over the flock animals, for example isolating a specified individual from the flock, or manoeuvreing them through a series of gates. Such displays of skill are the stuff of traditional country competitions and even the British television show 'One Man and His Dog'.

This work aims to demonstrate control of animal behaviour by reproducing just the central ability of the shepherd/sheepdog; **a purpose-built mobile robot gathers a flock of ducks and manoeuvres them safely to a specified goal position and holds them there indefinitely.**

The robot and flock operate in an otherwise empty circular arena to avoid the possible complications caused by corners and obstacles. The arena is as featureless as possible and the goal is chosen at random to prevent the ducks learning the task.

This work is concerned with producing behaviour in a mobile robot, such that its interaction with the behaviour of a group of animals achieves a useful task. As such it draws from past work describing the generation, control and modeling of behaviour from the fields of biology, agent-centered artificial intelligence and robot engineering.

2.2 Applications

Of course a robot sheepdog could be used to herd sheep. This is currently not considered seriously for pragmatic reasons outlined below (Section 2.5). However, an immediate application of this work is in turkey farming, where turkeys are raised in large barns with many hundreds of birds free to move around the floor. They must be collected for transport either to slaughter or to another farm. This requires human operators to enter the smelly, possibly unhygienic barns, collect groups of birds and chase them into a collecting area where they are put into crates. The turkeys are fearful of humans and run in a flock to avoid them, making the task difficult and unpleasant for both birds and operators. A machine using our proposed robot sheepdog technology could perform this task automatically.

Broiler chickens are raised in a similar environment to the turkeys. Broilers can not be herded, and are already harvested by machines, but are currently inspected twice a day by a human operator. The operator walks through the barn packed with hens observing their condition and behaviour, removing dead and sick hens. The broiler barn environment is unpleasant and the task is very monotonous. A robot inspector could move continually through the barn, observing the response of the surrounding hens as it moves. Hens that stay still or move atypically could be collected for human inspection. Work is underway on vision systems for tasks in broiler barns at Silsoe Research Institute in association with the University of Leeds.

Raising poultry is a major worldwide industry in which competition is strong and margins are tight. In 1995 865,000 tons of chicken meat and 183,103 tons of turkey meat was produced in the UK alone [Martin, 1997]. Automation has already made a significant contribution to processing poultry; there may yet be economic and welfare benefits for birds and operators in automated management of livestock.

Future possibilities...

A more fanciful possibility is the use of this project's flock control technology to ranch fish at sea. Robot submarines could herd schools of farmed fish out to sea for months at a time, bringing them back to shore for harvesting or inspection. Alternatively, submarine sheepdogs operating from fishing boats could dive to collect fish and drive them to the surface for netting.

The study of animal/machine interaction may provide insights useful when designing future machines to interact with humans. Robots working successfully amongst humans have the same requirements of safety and sensitivity to behaviour as for animals. Animals are comparatively naive subjects, with (often) simpler interactions with their environment, and it is hoped that animal-interactive experiments will provide a useful background for human-interactive robotics.

2.3 Previous work and background

2.3.1 Flocking in Animals

There is a huge literature related to the evolution of herd behaviour in real animals. A 28 year-old review article [Shaw, 1970] cites 61 articles about schooling in fishes alone. Trawling through the literature we find it generally accepted that herding offers its disciples various benefits for hazard- avoidance, foraging and mating [Pitcher and Parrish, 1993, Parrish, 1992, Caldwell, 1986, Ranta et al., 1993, Cresswell, 1994].

Studies vary on their emphasis; for example Ryer considers the advantages of herd life for communication [Ryer and Olla, 1991], but it could be argued that communication itself has evolved to support hazard-avoidance, foraging and reproduction. Once herding is adopted by a population, less-herding mutants are generally at a disadvantage and are selected against. Thus herding becomes an Evolutionarily Stable Strategy [Maynard-Smith, 1982].

What is flocking?

For a group of animals to count as a herd, the individuals must be actively seeking each other's proximity. Herding is considered an instance of *social organization* rather than a simple aggregation. Aggregations are groupings caused by environmental features, perhaps a food source or watering-hole, which attract numbers of individuals by common preference, rather than any social behaviour [McFarland, 1985, Hamilton, 1971, Martin and Bateson, 1993a]. Reynolds has defined flocking in flying birds as polarized, non-colliding, aggregate motion, where polarized refers to the common orientation of an animal group [Reynolds, 1987, p25].

For our purposes 'herd' is treated as synonymous with 'flock', 'school', etc.

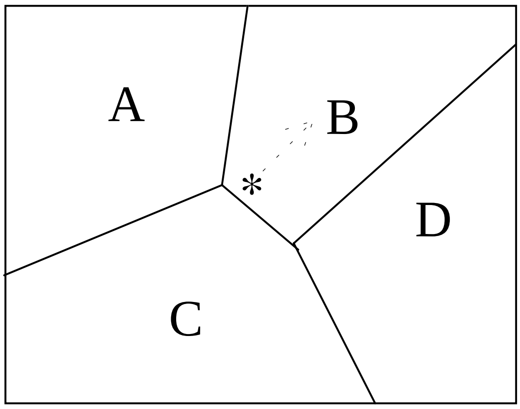


Figure 2.1: The 'domains of danger' for animats A-D. Predator '*' will attack the animat nearest to it; in this case B.

The Selfish Herd

In his paper 'Geometry for the Selfish Herd' [Hamilton, 1971] W.D. Hamilton describes a simple model of predator/prey interaction and proposes an anti-predator strategy for individual animals which is alone sufficient to produce group-level herding behaviour.

The paper was presented as an argument against group selection theories of evolution, and gives an abstract situation in which groups of animals will tend to form by the *uncoordinated selfish action of individuals.* In the model each animal seeks to reduce its chance of being eaten by an invisible predator by reducing its 'domain of danger'; that area containing all the points which are nearer to it than any other individual (Figure 2.1). Hamilton proposes a simple strategy: "...it must be a generally useful rule for a cow to approach its nearest neighbour. This is a rule for which natural selection could easily build the necessary instincts." [ibid, p304].

Flight Distance

The concept of flight distance is central to the literature on flocking. An animal's flight distance is that distance inside which the animal will move away from a perceived threat. In a reasonably low-stress situation an animal will move to approximately maintain this distance as a threat approaches. Figure 2.2 shows the uniformity of flight distance in a flock of sheep. This feature can be exploited to drive animals in a desired direction. This together with the tendency for herd animals to move close together in the presence of a threat forms the basis of flock control as demonstrated by shepherds through history. Figure 2.3 is an illustration taken from a book teaching flock control

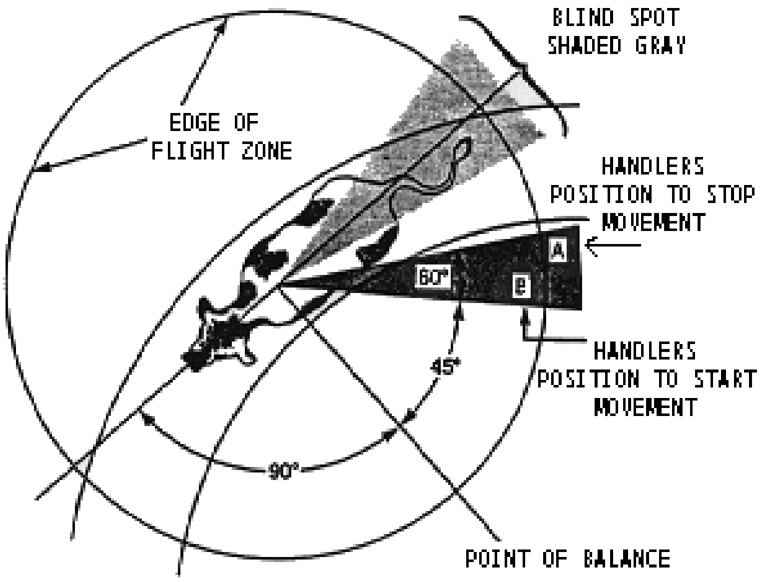


Figure 2.3: The flight zones of a cow, from Grandin [Grandin, 1989]

technique to farmers using such features as 'flight zone' (flight distance) and the 'point of balance' where the animal is motivated to move neither forward or backwards. [Grandin, 1989].

2.3.2 Flocking in Animats

Animat: a model animal

The term 'animat' was coined by Stuart Wilson to refer to the simulated animal-agent in his paper 'Knowledge Growth in an Artificial Animal' [Wilson, 1985]. The original animat was a simple agent moving in a structured cellular world, losing energy by 'moving' and gaining energy by 'eating'. Over time its classifier system 'brain' learned to exploit the structure of the world to increase its feeding efficiency.

Wilson's animat was designed to be a *minimal* model, where the simplicity helped to elucidate the behaviour. Recently animat models have become increasingly complex, to the point where their behaviour is very hard to analyse (see eg. [Werner and Dyer, 1992, Reynolds, 1994]). This reduces the lucidity and explanatory power of such models, and thus their value. Animat models in the Robot Sheepdog Project are deliberately minimal, hopefully keeping the spirit and power of the early animat work [Cliff and Bullock, 1993] and holding some interest for biology [Boekhorst and Hogeweg, 1994, Miller, 1994].

Animat has come to refer to any simulated *whole behaving agent*, especially those used to study learning and animal behaviour.

Animat Flock models

Several authors have written about group behaviour in animats, largely taking two approaches. The first is epitomized by the boids model, where agent behaviour is carefully programmed to result in spectacularly realistic flocks of animat birds.

Boids

The best known and most successful flock model was developed by animator Craig Reynolds, and presented in the journal 'Computer Graphics' [Reynolds, 1987]. It provided a means of automatically generating very realistic animations of flocks that would be very tedious to produce by hand or by previous computer graphics techniques. The most important feature of the model is that flocking is taken to arise by the *interaction* of a group of individually behaving agents. Each agent produces behaviour based on a local view of the world in a local coordinate frame with the agent at the origin. This agent-centered or *diectic* representation style is important in agent-based AI [Agre and Chapman, 1990] which includes most of the animat literature. Reynolds calls his simulated agents 'bird-oids' or *'boids'* and his flocking system a *distributed behavioural model.*

To simulate a flock, Reynolds simulates the physics of aerodynamic flight, "portions of the birds' perceptual mechanisms" and "that portion of a bird's behaviour that allows it to participate in a flock" [Reynolds, 1987, p25]. If the individual behaviour is right, he suggests, a population of boids running in the simulator should produce a flock through their interactions.

Simulating the physics of flight is straightforward; the interesting parts of the model are concerned with the boids perception and their behaviours. For perception, rather than simulating vision directly, boids are presented with"approximately the same information that is available to a real animal as the *end result* of its perceptual and cognitive processes". This amounts to a rather strong claim about the cognitive processes of birds. Each boid is supplied with the position, orientation and velocity of all the other boids within a "spherical zone of sensitivity" centred at its body. Those boids inside this zone are defined as its *neighbours.* The contribution of other boids to the behaviour of the centre boid varies with an inverse exponential of distance (as suggested by [Partridge, 1982] and [Warburton and Lazarus, 1991] discussed below). Thus the boid relies on perfect but artificially limited information about its environment. Other authors have developed flocking systems based on more biologically plausible simulated vision, see eg.[Renault et al., 1990].

The boids act according to these programmed behaviours:

1. Collision Avoidance: avoid collisions with nearby flock-mates

2. Velocity matching: attempt to match velocity with nearby flock-mates

3. Flock Centering: attempt to stay close to nearby flock-mates

Collision avoidance and velocity matching both serve to keep the boid on a clear course in the virtual sky. Collision avoidance causes the boid to steer away from nearby objects (boids and obstacles) to avoid hitting them. Velocity matching causes the boid to adjust its orientation and speed to that of its neighbours. This helps to avoid future collisions. Flock centreing causes the boid to steer towards the centre of its neighbours, aggregating the boids to form a flock.

Each behaviour generates an independent steering 'request' based on the state of the boid and its environment. These requests are arbitrated by a 'navigation module' which calculates a prioritized average steering command, which is then executed by a 'flight module'.

The three behaviours are attractively simple, and though the arbitration system Reynolds describes seems very artificial, the flock animations produced are very impressive. It will be shown in Chapter 3 that a much simplified model can produce satisfactory results, and it is hoped that further work will show that the simple model will well match natural flocking behaviour and generalize without the need for an arbitrary arbitration scheme.

Other models

The other main approach has been to study emergent mass behaviour in simple agents and robots, as practiced by Mataric and Maes [Mataric, 1992] [Maes and Brooks, 1990]. Mataric goes on to propose methods for acquiring behaviour that gives a net collective benefit [Mataric, 1994]; behaviour that is rather more 'social' than simply using one's sister as a shield from attack. Maes has compiled a good collection of early papers on the design of agents in general [Maes, 1990].

[Warburton and Lazarus, 1991] offers a simulation model of the 'forces' keeping social groups together . This examines the effects of various 'attraction/repulsion distance-functions' on the dynamics of the herd. Varying the function curves balancing mutual attraction and repulsion between members of a simulated population, they found that "all models led to stability in group structure, but differed significantly in terms of stable inter-individual distance and group structure"[ibid., abstract]. They claim most satisfaction with those models in which the attraction to a neighbour decreases with distance in accordance with the inverse-square law, as "movement tendencies are proportional to the size of a conspecific's image on the retina"[ibid. p485]. The efficacy of this type of direct correspondence between sense and action is demonstrated in the adaptive behaviour literature by Webb's work on phonotaxis in crickets and robots [Webb, 1993] [Webb, 1994, Webb and Hallam, 1996], and *in extremis* by the Braitenberg vehicles [Braitenberg, 1984].

Artificial Evolution of Flocking

Reynolds has evolved coordinated motion in cloned groups of animats [Reynolds, 1992]. A genetic programming method [Koza, 1992] is used to produce control systems. A candidate controller is copied to 20 visually guided animats placed in a world containing fixed obstacles and a pre-programmed predator. Animats 'die' if they touch an obstacle or the predator. The controller is scored over a fixed length trial according to an explicit fitness function based on the survival of the animats and some 'style' criteria designed by Reynolds to encourage attractive flocking [Reynolds, 1992, p388].

Reynolds is looking for (and gets) interesting coordinated motion. The influence of the predator on evolved behaviour is obscure due to the additional evolutionary pressure from a complex environment and the arbitrary nature of the fitness function.

[Werner and Dyer, 1992] also evolves herding behaviour, this time in an ecosystems model as advocated by e.g. [Wheeler and de Bourcier, 1994]. In this model animats are exposed to all the pressures understood to support herding. They must avoid predation, forage for food and find a mate. Here the pressure from predation is not constant, as the number of predators and animats is not fixed, but varies depending on their relative success. This plus the non-predatory pressures of foraging and mate finding obscure the effect of predation on group behaviour. Here, as with Reynolds' paper, no measurements indicating consistent behavioural trends were presented.

Zaera et.al. set out to evolve a flocking model along the lines of Reynolds' boids, but failed to find an effective evaluation function [Zaera et al., 1996]. They conclude that some problems (such as flocking) are hard to express in an evaluation function; a problem for engineering based on simulated evolution. The difficulties of such techniques were described in detail in [Mataric and Cliff, 1996]. This author has demonstrated the artificial evolution of flocking behaviour *without* an evaluation function: an ecosystems style system avoids the problems Zaera encountered in this case. This work will be submitted for publication elsewhere.

False flocking

It is quite possible to mistakenly interpret mass non-social behaviour as herding by means of emergent aggregating effects. An agent based model of foraging patterns in orang-utans found

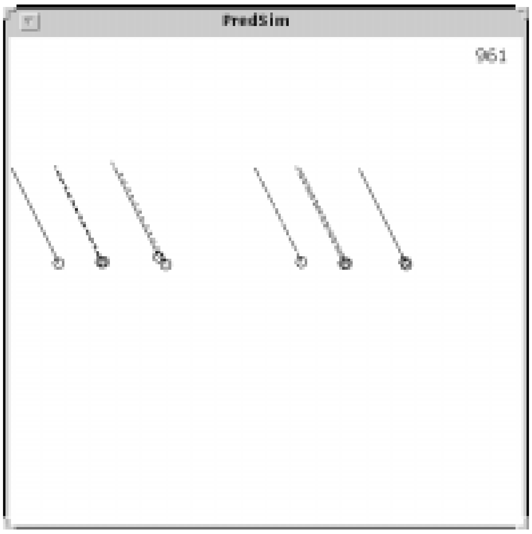


Figure 2.4: Apparent but false flocking in an animat simulation. Animats (circles) leave trails for 10 timesteps to indicate direction and speed.

that the location and distribution of food sources in the environment produced an aggregation of otherwise antisocial orang-utanimats traveling together, a finding supported by observed field data [Boekhorst and Hogeweg, 1994].

A related phenomenon in A-life systems is the possibility of structured, systematic behaviour from a number of randomly generated, but identical, animats. Often the system will relax into a regular pattern; for example Figure 2.4 shows a common equilibrium for some simple animats when given copies of a *purely random* control system. The human tendency to apply the Argument from Design in the face of apparent systematicity can lead to problems of interpretation, here as elsewhere.

Boids for biology

In the original boids paper Reynolds considered the usefulness of his model to biologists:

> One serious application would be to aid in the scientific investigation of flocks, herds, and schools. These scientists must work almost exclusively in the observational mode; experiments with natural flocks are difficult to perform and are likely to disturb the behaviours under study. It might be possible, using a more carefully crafted model of the more realistic behaviour of a certain species of bird, to perform controlled and repeatable experiments with "simulated natural flocks." A theory of flock organization

Figure 2.5: The University of Western Australia's Robotic Sheep Shearer.

> can be unambiguously tested by implementing a distributed behavioural model and simply comparing the aggregate motion of the simulated flock with the natural one. [Reynolds, 1987, p32].

Some experimenters have used flock models as Reynolds suggests, examining animal systems such as distributions and movement patterns in schools of fish [Nepomnyashchikh and Gremyatchikh, 1996],

Rather than performing ethology experiments with a simulated flock as Reynolds suggests, this thesis describes how a simple flock model can be used as an engineering tool, to aid the design of a real-world flock-controlling robot.

2.3.3 Robots and animals

Few extant robot systems are concerned at all with (live) animals. Those that do are almost all concerned with avoiding them, eg. *Herbert* the can-collecting robot which avoids bumping into human office workers [Connell, 1990]. Robot-like machines designed to have a direct effect on humans include medical tools (largely tele-operational, eg. [Arai et al., 1996]) and aids for the disabled (eg. [Yukawa et al., 1996]).

Those concerned with non-human animals include the robot sheep shearer developed at the University of Western Australia [Trevelyan et al., 1983, Trevelyan, 1989, Trevelyan, 1992], and Silsoe

Research Institute's dairy cow milking robot [Frost et al., 1993]. The sheep shearing system incorporates a special restrainer to minimize the sheep's movement while the robot shears the fleece. The restrainer was so successful that it is being developed commercially for use by human shearers. The milking robot is part of a larger system which exploits cow behaviour. The cows come into the milking parlour when they choose, and are milked and fed without human intervention. This gives advantages in welfare and (potentially) labour efficiency [Prescott, 1995]. Once in the parlour though, the cow is constrained in a stall while the robot attaches the milking equipment.Neither of these systems are interactive in any strong sense. Both rely on keeping the animal as still and non-reactive as possible while the robot does its job.

The main areas of research for robot[1]/human *interaction* are in human telecommunications and entertainment. Neither of these areas are much concerned with *physical* interaction, but are rather *information-interactive*, and therefore of little use for work with non-human animals.

Some authors have used robots as models of animals. For example Barbara Webb has built robots to demonstrate mechanisms for perception and motor control in crickets. Experiments suggested that females use a simple direct sense/action control scheme to orientate themselves when finding a mate. Crudely put, the crickets' ears are wired directly to their leg muscles so that the chirping of a male, louder in one ear then the other , causes the female to turn towards the sound. Webb implemented this controller in a robot, and found that it could reliably approach a 'mate', even in the presence of noise [Webb, 1993].

David McFarland compares robot and animal design; in particular he suggests that behaviour is organized according to economic principles [McFarland and Bosser, 1993] and has produced robot models that demonstrate effective behaviour-selection along these lines [Spier and McFarland, 1996]. McFarland is among those who suggest that robot designers could look to animals for design principles, and that ethologists could use robots to test hypotheses about behavioural mechanisms (see eg. [Hallam and Hayes, 1994, Hallam and Hallam, 1994, Miller, 1994, Wheeler and de Bourcier, 1994, Anderson and Donath, 1990, Kleiner, 1994] for advocacy of this position). An example of a possible animal/robot design principle may seen in be the *independent* evolution of round, foveated eyes

[1] 'robot' in its broad sense of autonomous intelligent agent

in several species. Eyes with this arrangement have powerful properties[2] that can reduce processing requirements. David Young has shown that such a biologically-inspired vision system can do image matching tasks very efficiently compared to conventionally engineered systems [Young, 1988].

2.3.4 Behavioural Robotics

There is a paradigm of robot research which considers behaviour to be central to the study and design of intelligent agents. This thesis describes the production of behaviour in order to control existing animal behaviour, so it is particularly relevant here. The originator and champion of *Behaviour-based* or (later) *Behavioural Robotics*, Rodney Brooks, lays out the "key aspects characterizing this style of work

- *Situatedness:* The robots are situated in the world - they do not deal with abstract descriptions, but with the here and now of the world directly influencing the behaviour of the system.
- *Embodiment:* The robots have bodies and experience the world directly–their actions are part of a dynamic with the world and have immediate feedback on their own sensations.
- *Intelligence:* They are observed to be intelligent - but the source of intelligence is not limited to just the computational engine. It also comes from the situation in the world, the signal transformations within the sensors and the physical coupling of the robot in the world.
- *Emergence:* The intelligence of the system emerges from the system's interaction with the world and from sometimes indirect interactions between its components - it is sometimes hard to point to one event or place within the system and say that is why some external action was manifested" [Brooks, 1991].

2.4 Approach: Fresh AIR

As far as the author can see, the Robot Sheepdog Project is the first robot system that is designed to truly *interact* with animals. It is argued below that such a system is *necessarily* a subset of Behavioural Robotics. This distinct research area will be referred to as Animal Interactive Robotics (AIR).

[2] Rotation and expansion of images on the retina correspond to simple 2-D translations in image space.

2.4.1 Statement of methodology

The methodology of Animal-Interactive Robotics developed in this work is to exploit reliable behaviour by means of appropriate interaction.

A *reliable behaviour* is one which can be exploited to achieve a useful task. This thesis demonstrates that the threat-avoidance behaviour in flocking ducks is a reliable behaviour.

An *appropriate interaction* is a behaviour which, when combined with an existing reliable behaviour, achieves a useful task.

A reliable behaviour and an appropriate interaction combine to produce a new reliable behaviour. By definition, the new reliable behaviour can be exploited in turn to achieve a further task. Thus the methodology allows scaling by adding layers of appropriate interaction.

2.4.2 Designing interactions with animats

In many cases, once the reliable behaviour has been identified, designing the interaction is straightforward. For example the reliable behaviour of many birds is to flee for cover following a loud noise. The farmer exploits this behaviour at sowing time by installing gas guns to regularly scare off the birds with a loud bang. Many insects are reliably attracted to ultraviolet lights; the baker installs such a light to keep them away from his buns. The robot milking machine [Frost et al., 1993] exploits the reliable behaviour of dairy cows which will, once they have learned that the machine relieves them of their milk and that food will follow, come into the robotic parlour every day of their own accord. These reliable behaviours - hazard avoidance, light-referenced navigation, and the need to empty udders are exploited to human benefit by placing machines in the animals' environment.

The interaction required for a robot sheepdog is altogether more sophisticated as it incorporates *feedback* between the animal and the interacting agent. New models of animal behaviour, particularly the animat-style behavioural simulations, may facilitate the design of more complex interactions. Such models provide a simulation tool with the advantages enjoyed by conventional engineering simulations in terms of convenience, cost, repeatability, etc. Indeed, simulations of crowd movement have already been used to design public buildings (at least one commercial product exists; "Rampage" by Animation Science [http://www.anisci.com/RAMPAGE/rampage1.htm]. In addition, the process of building the model in the first place may provide important insights into how the behaviour may be exploited. It is suggested that animal-interactive robotic systems should be designed at the same level of abstraction as these behavioural simulations.

It is further suggested that reliable behaviours lend themselves to modeling, in that they tend to be relatively simple and can be isolated from the rest of the animals' behaviour. The behaviours are reliable for the designer because they have to be robust for the animal, eg. to prevent them from being eaten. They achieve robustness through simplicity. They can be isolated because they are performed in isolation; when performing a reliable behaviour an animal will not perform other behaviours.

These ideas are demonstrated in this thesis by using a simple model of flocking behaviour to facilitate the design of a robot sheepdog. The robot controller is constructed from the same 'toolbox' as the flock model; the same mechanism that *describes* the flock motion is used to *generate* the motion of the robot.

2.5 A Robot Sheepdog: Fowl AIR

Sheepdog vs. Duckdog

The engineering problems of building a robot with the physical capabilities of a real sheepdog are beyond the state of the art and the scope of this project. Herding sheep can require great speed and agility on a range of terrains. The emphasis in this work is on the animal/machine interactions and the mechanical engineering is kept as simple as possible. With this in mind, domestic ducks were chosen as the subject flock animals. Human shepherds recognize that duck flocking behaviour is very similar to that of sheep, and can be controlled by a sheepdog. Indeed, ducks are often used by trainers to refine a dog's delicate close-quarter herding skills (Figure 2.6). Ducks are a major source of meat in Asia and are driven around the farm in much the same way as sheep (Figure 2.7). They are much smaller and slower than sheep, thus simplifying the experimental requirements considerably. They are also more economical to buy and keep. The term 'sheepdog' continues to be used because 'duckdog' doesn't have the same explanatory power.

2.5.1 Rover as Behavioural Robotics

Let us characterize the RSP system in terms of Brooks' criteria for Behavioural Robotics:

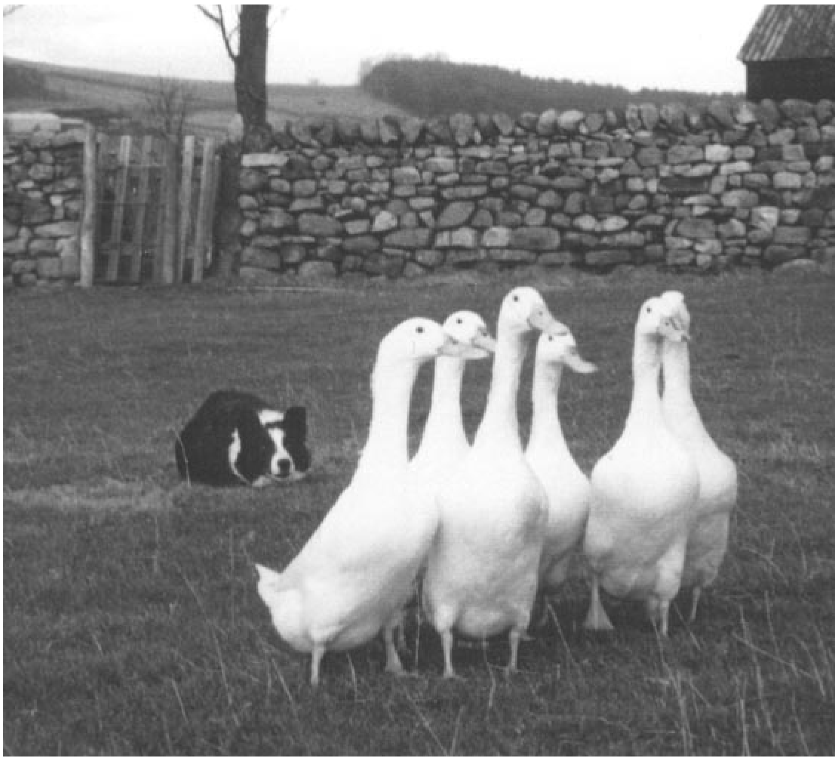


Figure 2.6: A young sheepdog in training with a group of ducks. Photographed by the author in Lancashire, November 1995.

Figure 2.7: Indonesian duck-herd with his flock.

- Rover is *situated*: its actions are determined by sensing the actions of independent external agents, whose behaviour is not explicitly modeled in the controller.
- Rover is *embodied*: it has a physical vehicle which is used to affect changes in the world in human (and animal) time scales. Its behaviour has an immediate and direct effect on the world.
- Rover is *intelligent*: it does a job which is considered to require intelligence in humans and animals [Boden, 1987]. Its efficacy depends on its correct ongoing interaction with the complex dynamic world.
- Rover's effect on the world is *emergent*: the goal of its actions is predetermined, but its trajectories are not prescribed entirely by its controller. Rather they emerge from interaction with unpredictable external agents. Rover's behaviour is *meaningless* except with respect to the behaviour of the flock.

Any animal-interactive robotics system will score on all these points by the nature of animal interaction, independent of its own internal architecture. Brooks would argue that such a system is likely to be unsuccessful unless it is designed along behavioural principles. Rover's controller is designed with these issues firmly in mind, though not using a typical Brooksian architecture (see [Brooks, 1986] for a description of Brooks' favoured *subsumption architecture*). A potential-field method closely related to that which drives the flock simulation produces a fast, simple and effective controller. Potential-field controllers are fairly common in mobile robotics and have advantages for robustness and flexibility (see [Cameron and Probert, 1994] Chapter 11 for a good overview). As the flock control problem is about the relationship between the robot and the flock, it seems appropriate to try to construct both from the same building blocks.

2.5.2 Summary of methodology

1. Identify a reliable behaviour that you mean to control
2. Develop a simulation model of the behaviour, based on existing behavioural studies or new ones if necessary
3. Design an appropriate interaction which controls the behaviour in the simulation

4. Test the interaction with the real animals.

This specific methodology is a subset of the general behavioural robotics approach of "exploiting system dynamics to do the work for you" [Mataric, 1998, personal communication].

These methodological steps turn into specific hypotheses when applied to the example RSP task:

2.5.3 Hypotheses

1. Robotic flock control can be achieved by exploiting the flock animals' threat-avoidance behaviour.

2. The appropriate interaction is to position the robot behind the flock with respect to the goal while maintaining an appropriate robot-flock distance.

3. A simulated flock could be used to design and test a robot controller that achieves (2).

The rest of the thesis examines these hypotheses.

2.6 Summary

An overview of the literature on flocking was presented. The animat, flock modeling and work on artificial evolution of flocking were described. The Boids model of flocking as a decentralized activity was described, and is suggested as a useful tool for the study of natural flocks. A brief discussion of relevant robot research covered previous work in both animal *inspired* robotics and robotic *models* of animals. The field of *Animal Interactive Robotics* was identified, of which the Robot Sheepdog Project is the first example. AIR was placed in the context of Behavioural Robotics. The methodology used to develop the robot was described and specific hypotheses were stated, to be tested in the experiments in this thesis.

The next chapter describes the development of the project's robot vehicle.

Chapter 3

Rover the Robot Sheepdog

This chapter describes the RSP experimental set-up; the systems, hardware and software that were developed to enable the experiments in this thesis.

3.1 Experimental design

The experimental goal of this work, as described and motivated in Chapter 2 was to demonstrate control of animal behaviour by designing a mobile robot to interact appropriately with a flock of ducks. The equipment and procedures described here were designed as the minimum system required to achieve this goal.

3.1.1 The RSP distributed system

Naturally, the real shepherd and sheepdog team was the first place to look for inspiration. A shepherd and sheepdog form a distributed system. In general the shepherd has a global view of the situation, including knowledge of the goal state. He can control the behaviour of a dog with a few whistled commands, meaning (roughly) 'towards me', 'away from me', 'round to the left', 'round to the right', 'faster', and 'slower'. Thus the shepherd makes (possibly complex) control decisions based on his global view and communicates them to the dog via a very low-bandwidth medium (the whistles). The dog interprets the command in the context of its own local state and executes the detailed muscle actions, etc., to produce the indicated movement. The shepherd need have no idea how this works, as long as he can predict reasonably accurately what the results will be. The shepherd can predict the outcome of a particular dog movement based on his experience of similar past interactions.

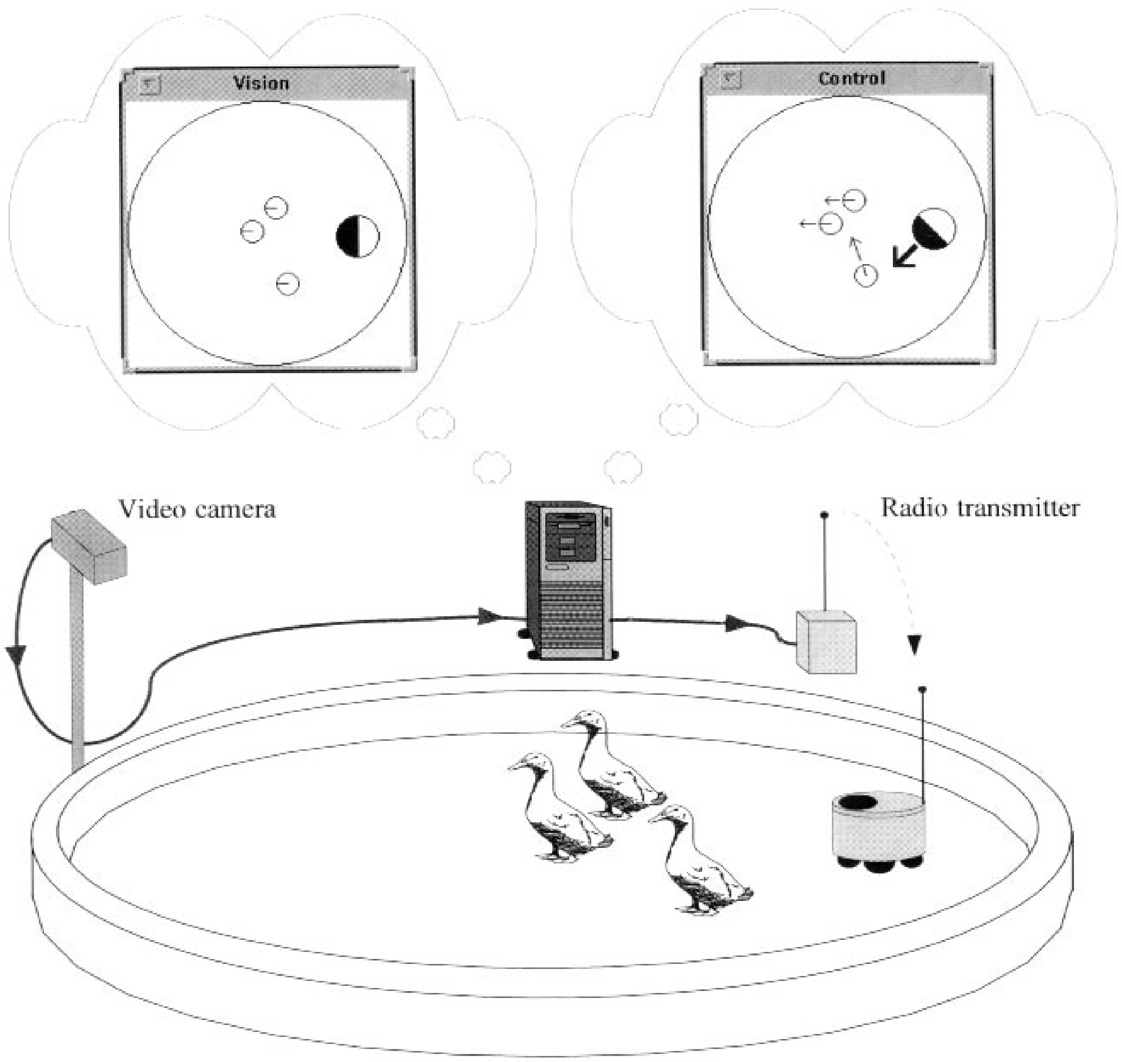


Figure 3.1: Robot Sheepdog system layout.

This general arrangement is reproduced in a robot system, subject to the caveats below. Figure 3.1 shows the overall system design. A video camera provides a global overhead view of the whole arena; a computer decides the appropriate behaviour for the vehicle; and a vehicle executes that behaviour. This system was used as a platform for the experiments in flock control described in Chapters 4 and 5.

3.1.2 Scope and limitations

It was determined right at the start of the project that no attempt would be made to reproduce the physical abilities of a real sheepdog. It is assumed that the problems of moving rapidly over uneven terrain are general, separate engineering issues. Likewise issues of long-term autonomy and robustness. Solutions to these problems would be necessary but not sufficient to allow the construction of a useful Robot Sheepdog. This work examines the kind of *control strategies* that must be developed if a robot is to gather animals. Thus this work and the Project as whole focuses on the nature of the *interaction* between the flock animals and the dog/robot.

There is no vision onboard the vehicle. It is recognized that the lack of vision onboard the robot vehicle is a major difference between the real shepherd/sheepdog system and this simplified robot version. The information available to the vehicle (sheepdog) is greatly reduced and therefore requires the high-level offboard controller (shepherd) to guide the vehicle in much more detail than in the real system. The nature of the problem remains essentially the same.

Similarly, it is recognized that dogs are capable of herding sheep autonomously, without guidance from a shepherd, using only their own 'on-board' sensors. Several reviewers have suggested that reproducing this ability on a robot would be more interesting than our distributed system. Chapter 6 demonstrates exactly this in simulation. Transferring this ability into the real world would be a significant challenge and was unfortunately beyond the time and budget constraints of this project. The algorithms used to control the simulated robots with onboard sensing are very closely related to those demonstrated on the real robot using offboard sensing.

At the beginning of the project, pilot trials were conducted outdoors in a large (15m) grass arena. The robot vehicle was designed at this stage to work in this environment. However, the outdoor arena was abandoned after it became apparent that it would be difficult to build a reliable vision system that would work outside. Rather than devote a great deal of time to the vision system and away from flock control, it was decided to move indoors into a more controlled visual environment. Because this constraint was pragmatically rather than theoretically motivated, it was ensured that nothing (apart from the vision system) *relies* on the arena being indoors, but should work happily outside in a large arena. This explains why the vehicle may appear over-engineered for the final experimental environment.

3.1.3 System components

We identified these elements to be required for a robot sheepdog system:

1. *a suitable experimental arena and flock*
2. *a vehicle to interact with the ducks*
3. *a means of determining the positions of the ducks and the robot*
4. *a model of the ducks' response to the vehicle, to be used to design*
5. *an algorithm to control the vehicle and effectively herd the ducks*

The rest of this chapter describes each of the first four elements in turn, while the next two chapters each introduce a candidate flock-control algorithm.

3.2 Arena

The RSP has its own workshop which houses the ducks, an experimental arena and the computing equipment. The arena is 7m diameter with 2m high wooden walls and a concrete floor and is shown in Figure 3.2. This is the largest arena that could be built in the workshop. Some wall panels are detachable for ease of access to the arena. Two panels can be raised by a system of ropes and pulleys; these remotely opened doors allow the ducks in and out without disturbing them by human contact. The whole arena is painted a plain yellow-brown colour; Jane Henderson required that the arena be as visually neutral as possible for her experiments. The arena was built largely by Jane and BED's craftsman Len Burgess.

A monochrome CCD video camera is mounted in the roof above the arena, as close as possible to the arena center.

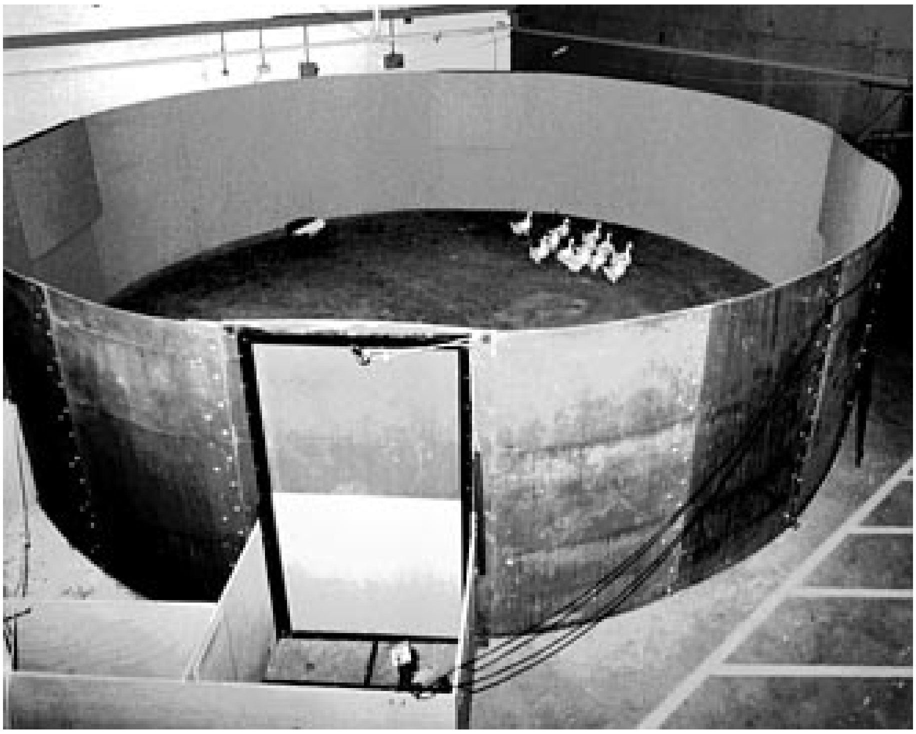


Figure 3.2: The experimental arena

The ducks are white and easy to see in video pictures, and are a meat-producing breed which means they are relatively slow moving and well-behaved (Figure 3.3. The number and type of birds used in experiments are described more exactly in the experimental chapters 4 and 5.

3.3 Robot

An existing vehicle could not be found to match the specification, so a purpose-built robot sheepdog was constructed. In the tradition of mobile robotics it is called Rover. This section describes Rover's design and construction.

3.3.1 Labour summary

The robot was specified and designed by the author and the support staff at SRI. The author built a working strip-metal prototype which was the basis for Steve Crook's strong aluminium chassis. John Lowe designed and built the custom microprocessor board and modem and also coded some of low-level assembly routines. Paul Griffith designed and built the analogue PWM speed controllers and made the box and some of the cables.

Initial design, component selection and aquisition, most low-level assembly and C programming

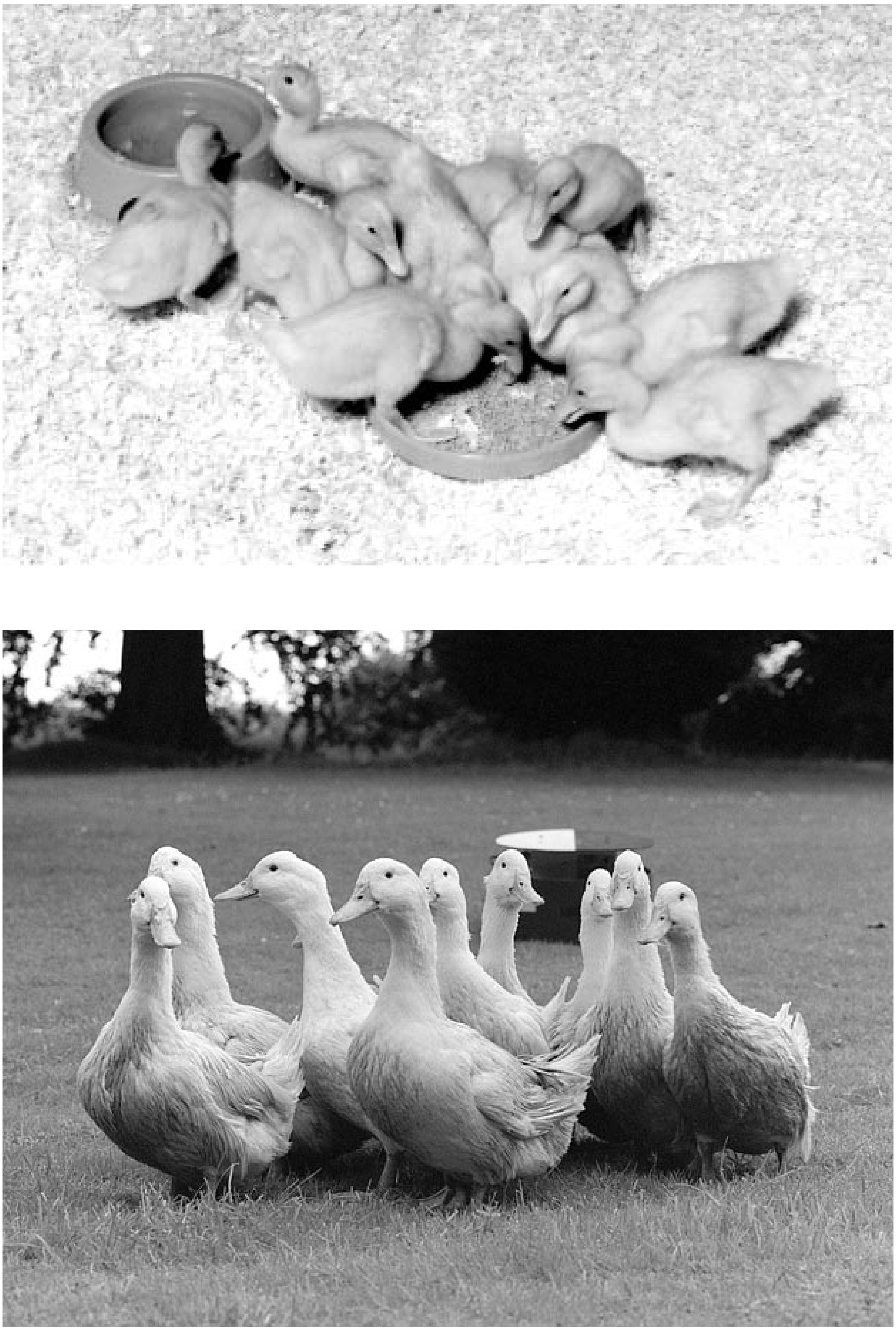


Figure 3.3: An duck flock raised at SRI and used for these experiments, around 5 days old (top) and adult (bottom) with the robot in the background.

of the microprocessor board, all of the control, communications, modelling, logging and visualisation software was completed by the author. System integration and debugging was the most time-consuming and laborious of the author's tasks. Robert Smith provided valuable assistance in the debugging process. Most components went through several iteractions and the robot took around 20 months to complete.

3.4 Design

The robot sheepdog system comprises a custom-built vehicle with onboard microcontroller and an offboard 200MHz Pentium-based Linux workstation. The vehicle and the ducks operate in an the arena, constantly in view of the overhead video camera. The workstation is equipped with a Matrox Meteor video digitising card which samples the camera images which are then processed to find the position and orientation of the robot and flock (see section 3.5 below). This image processing was likely to be most computationally intensive part of the process and the system was designed around an estimated vision update rate of 10Hz. The finished system achieves approximately 15Hz and the image processing takes up more than 90% of the processing time.

A 'strategic' control program also runs on the workstation. This program takes as input the position and orientation information from the vision system, and produces desired trajectories for the vehicle. The resolution of the vision data is poor in control terms (position to $\approx \pm 5$cm, heading to $\approx \pm 5^o$). This is not so important for determining the relative states of the flock and robot for analysis, but is a severe limitation when trying to control the vehicle, and means that a 'classical' control loop cannot be completed through the workstation. Instead the strategic control program outputs the required wheel speeds to the vehicle over a radio modem link and the vehicle is responsible for controlling its own wheel speeds. The modem has a relatively low transmission speed ($\approx$1KB/s), and is potentially a bottleneck for the system. Establishing a reliable high-speed radio link from this vehicle to this workstation would have been difficult (or at least expensive; radio ethernet was an emerging technology in 1995), so the design deliberately minimizes the communication necessary at this link in a way loosely analogous to the shepherd whistling to his sheepdog.

The vehicle itself has no measure of position or heading; these are observed by the vision system

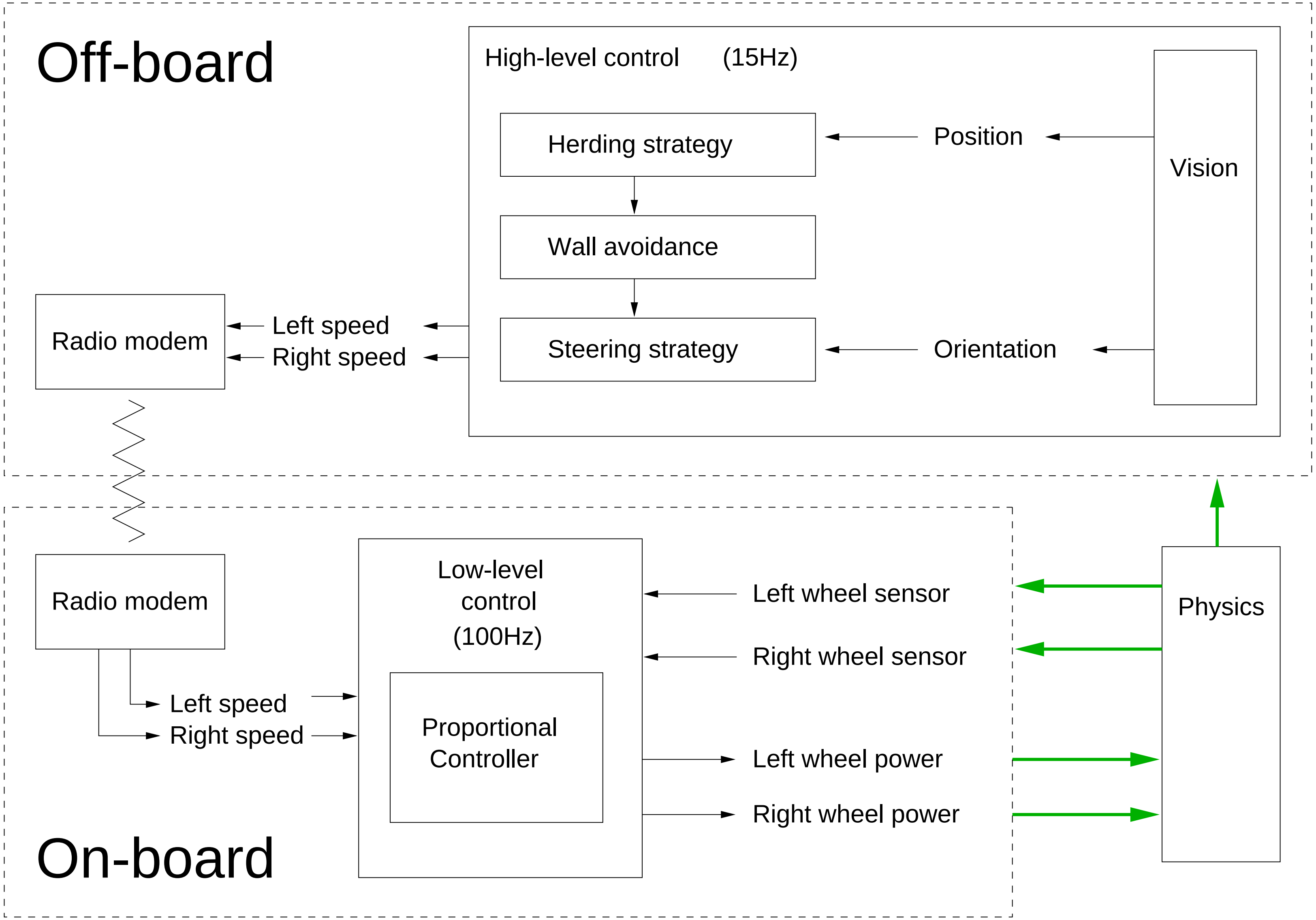


Figure 3.4: Control schematic

and used to update the strategic controller's model of the world.

The vehicle has its own microprocessor-based controller, which receives the desired speed and turn rate, calculates the necessary wheel speeds, compares them to its own current wheel speed measurements, and adjusts the power to the wheels appropriately. The vehicle then moves according to the vehicle/terrain physics, the ducks move according to the influence of their environment, and the new positions are observed by the camera.

Factors such as wheel-slip, uneven terrain, and poor orientation estimates mean that the vehicle will never quite achieve the strategic controller's goal trajectories. But given the relatively large error in the estimate of flock position, and as the vehicle will influence the ducks by their relative, rather than absolute, positions, this is acceptable. The strategic controller continuously updates its desired trajectories in response to movement of the robot *and* the flock once per update cycle (around 15Hz).

3.4.1 Requirements

Mechanical

The vehicle is required to work in a duck's environment: outdoors, on short grass, and in real time. From the experience gained in the pilot trials it was decided that the vehicle should have a top speed of around 4ms^{-1} and acceleration of $\pm 2\text{ms}^{-2}$, which is about twice as fast as the ducks (see [Henderson, 1999] for data on duck locomotion). It should also have a good turn rate, and the battery life must be sufficient to allow uninterrupted trials with the ducks of around 20 minutes. To keep the the manoeuvrability and battery life as high as possible, the vehicle's weight should be kept down. A lighter vehicle would also be less dangerous to the ducks. The vehicle should have a cover that would minimize the chance of harm to the ducks in a collision.

Control

The vehicle should be able to receive wheel speed demands by radio from the workstation, achieve those wheel speeds quickly and maintain them until the next wheel speed demand is received. The vehicle should be able to receive new demands at least five times a second.

3.4.2 Design and Construction

The pilot trials showed that moving the flock using a radio controlled car was very difficult. The steering geometry of the car appeared to be a major limitation; control of the ducks required better manoeuvrability over small distances than the car geometry allowed.

A suitable design was identified, with two independently powered wheels on concentric axles. This configuration is frequently used in research robots as it allows for holonomic control and thus maximum manoeuvrability and offers simplicity of construction. This allows the construction of a circular robot which can turn on the spot, rotating about its centre. Thus the robot can turn and move off in any direction while presenting a similar visual stimulus to the ducks from any angle. It also permits simplifying assumptions to be made for some control strategies, is simple to simulate (see below), and minimizes the chance of the robot colliding whilst turning (see eg. [Cameron and Probert, 1994] for the advantages of holonomic control).

Rover #1

Figure 3.5 shows the first Rover prototype, built to asses the chosen geometry and first choices of wheels, motors and gearboxes. Parts chosen at this stage were all simple, affordable and easily available. Two standard '540' electric motors are used, as found in model racing cars, powered by a single 6V 20AH dry cell lead acid battery. Small lightweight gearboxes provided a 25:1 reduction ratio. The driven wheels are 10cm diameter plastic with rubber tread. For stability there are two castor wheels, which are somewhat smaller and raised sightly with respect to the driven wheels, so maximizing contact of the driven wheels with the ground.

These components were built onto a Meccano (strip-metal) chassis, along with with commercial radio and speed controllers. The prototype was driven manually around the original grass arena to test its ability to move on the terrain. The design was largely successful, with sufficient ground clearance, wheel size and grip, etc.

The chosen gearboxes proved to be difficult to mount and fragile in use, so these needed to be upgraded. The unreliable gearboxes meant that the power output and battery life could not be properly assessed. The Meccano chassis was not sturdy enough and had to be improved.

Rover #2

The second prototype (3.6) was based on a tough aluminium chassis, with larger all-metal gearboxes replacing the original models. The gear ratio was increased to 30:1 to (approximately) compensate for the increased weight of the vehicle. All the other components were the same as for Rover #1.

Rover #2 was tested by having the author drive it manually over the radio link. In this way Rover #2 successfully herded a group of 6 ducks around a circular pen. It was observed that (at least under manual control):

- The ducks moved away from the approaching vehicle;
- The ability to turn on the spot meant that the ducks were much easier to control than with the model car used in the original pilot trials;
- Careful low-speed movements were used for around 80% of the trial, with our top speed only rarely required.

The second remote-controlled prototype showed that the physical configuration was good and the mechanical performance exceeded the specification. The new gearboxes proved to be very robust and in combination with the original motors produced a top speed and acceleration in line with the specification.

The commercial speed controllers offered poor proportional control; it was difficult to control the vehicle at low speeds, where the trials suggested most of the work would be done. They also had a tendency to catch fire when run in reverse. Improved speed controllers were required, and these were custom-built to allow easy interfacing with the anticipated microcontroller.

Rover #3

The custom speed controllers are a dramatic improvement over the standard versions; Rover can creep along at 0.1 ms^{-1} or sprint it at over $4ms^{-1}$. Acceleration is smooth, braking effective by applying reverse power, and manual control is very easy compared with the previous vehicle. It can achieve a fast turn rate of up to 50rad/s (8 revolutions/s).

Rover #3 (Figure 3.7) was the first vehicle to be equipped with a microcontroller that could regulate the speeds of the wheels according to commands transmitted by radio modem from the

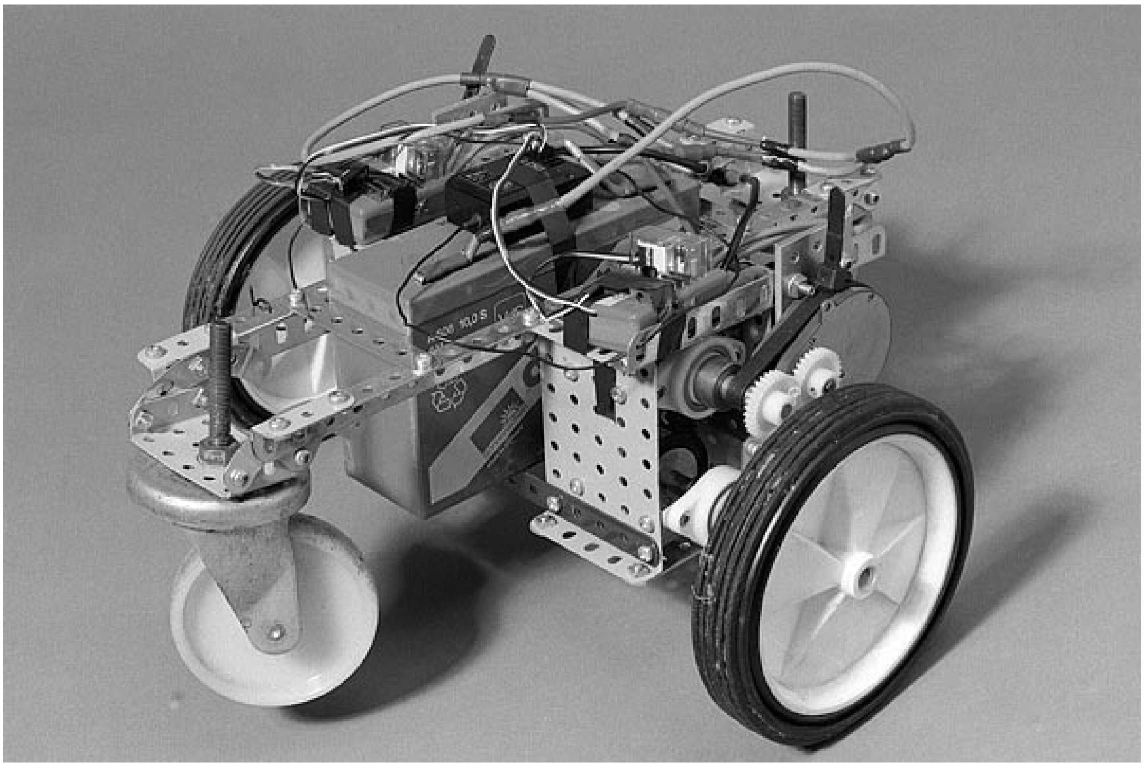


Figure 3.5: Rover #1: meccano prototype to test geometry, wheels, gearboxes and motors.

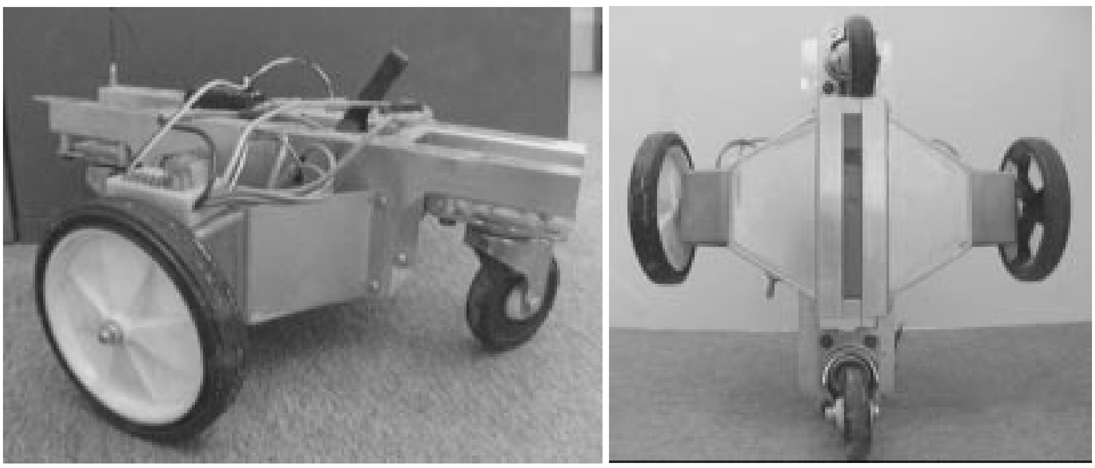


Figure 3.6: Rover #2: second prototype with custom chassis and sturdy gearboxes.

workstation. The chassis was modified to accommodate an electronics package and its accompanying battery, plus a radio aerial and base-plate to reduce interference (poor reliability plagued the radio modem at this stage) . The electronics and software described below were mounted on this vehicle. A cover was also fitted for the first time.

Rover #4

Following further trials with the ducks, including experiments conducted with Jane Henderson (reported in [Henderson, 1999]), it was concluded that the ducks would reliably move away from the robot, regardless of its appearance. The meant that the tall cover could be shortened, bringing the lid markings into view of the ducks, without any noticeable effect on their behaviour. This made Rover more stable (less top-heavy) and reduced the deflection of the lid as Rover shifted its weight from one castor to the other. This reduced the error in the position estimation from the vision system (see below). Problems with the reliability of the radio modem were painstakingly traced to a bug in the standard I/O library provided with the system. Once this was fixed it was found that the large base-plate for the aerial could be removed. The result was a much more compact, convenient cover. Rover #4 met or bettered the specification and has proved tough and reliable since then.

This is the final version of the vehicle and was used for all the real-world experiments in this thesis. Rover #4 is shown in Figure 3.7.

3.4.3 On-board electronics

A waterproof box houses a controller board, radio modem and high quality pulse-width modulated (PWM) speed controllers. The radio modem operates at 9600 baud, over a range of around 100m using a small aerial. A copper plate is fixed between the aerial and the electronics package to minimize radio interference.

The controller board provides a TMS 370C micro-controller running at 16MHz, D/A and A/D converters, 40KB memory and a real-time clock. Low-level assembly-language software for this controller provides routines to drive the radio modem, handles I/O for the D/A and A/D converters, and count the pulses from the wheel sensors. Commercial optical shaft encoders are used, each producing 1000 TTL pulses per revolution. Each pulse increments a running total

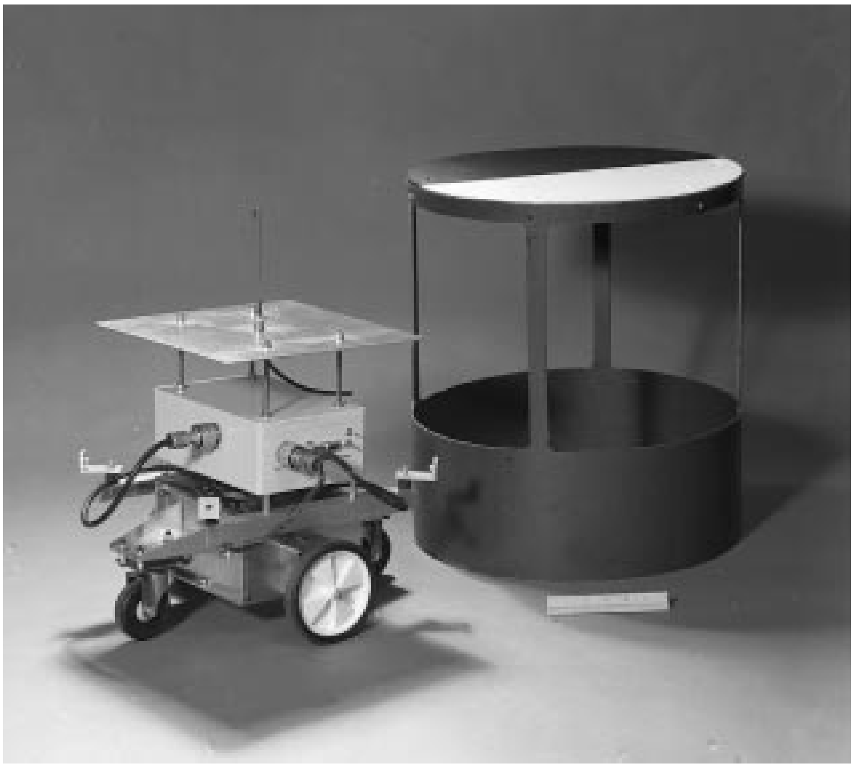


Figure 3.7: Rover #3: final chassis with radio base-plate and original tall cover (removed for clarity).

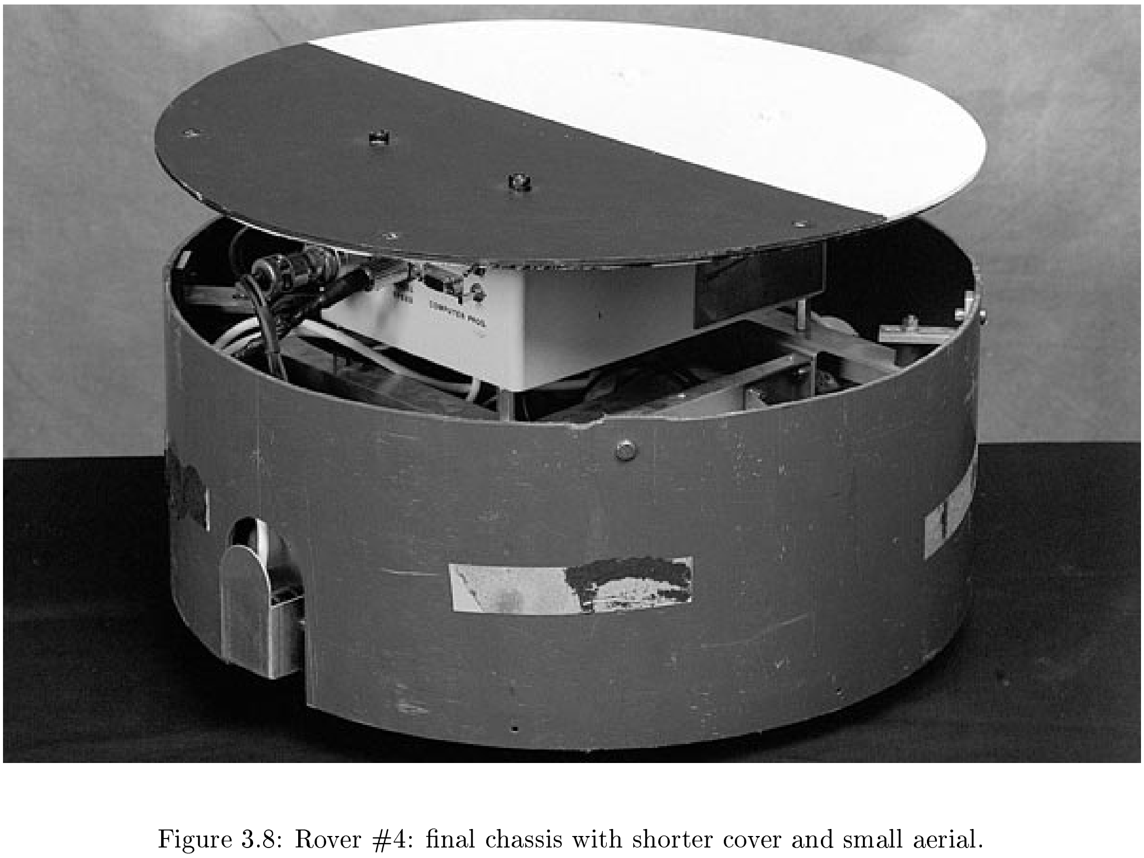


Figure 3.8: Rover #4: final chassis with shorter cover and small aerial.

maintained by a counter chip. The counter is interrogated by the microprocessor which compares the current and previous counter readings with the real-time clock to determine the wheel speeds. The duty cycle of the PWM speed controllers is set by software in 0.5% increments, providing fine-grained proportional control.

3.4.4 Radio communications

The strategic controller communicates with the vehicle through the radio modem. Short coded messages (a few bytes of ASCII) can be sent in both directions. The modem's maximum transmit rate of 9600 baud was reduced somewhat in practice due to radio interference, but was still comfortably fast enough to update the vehicle at 25Hz, the maximum anticipated update speed of the offboard controller (imposed by the incoming 25Hz standard video signal).

There are three types of command message sent from the strategic controller to the vehicle: the SPEEDSET message is the most used as it sets the vehicle's wheel speed demands. The CONTROLSET message sets the vehicle's on-board parameters, such as the proportional controller gain. In practice these are set once at start-up, but the ability to update these values on the fly gave scope for incorporating an adaptive controller. Such a scheme could update control parameters during a run to compensate for changes in the vehicle characteristics over time (due to heat, battery charge, etc.). The BASIC message sets the vehicle's low-level state directly. This includes the motor output level, real-time clock value, etc. This is used mostly for test and debugging, but it also lets us stop the vehicle in an emergency by cutting the power to its motors. When the vehicle receives a BASIC message it sends back a REPLY message which contains information about its current state. This is only used for testing, and never for control purposes.

3.4.5 Casing

A plain grey cylindrical cover is fitted, cut from tough plastic tubing and fixed on rubber mounts. This provides an effective bumper, protecting the vehicle and ducks from collisions. The original cover was 80 cm tall in order to be above the head height of our ducks ($\approx$0.7m). This allowed the lid to be designed to best suit the vision system, without effecting the duck's-eye view. Panels were cut out to save weight and covered in opaque plastic film to give as neutral a basic appearance as possible. This precaution turned out to be unnecessary, however, and the cover height was reduced as described above. Figure 3.7 shows Rover's original tall cover in the background. Figure 3.8 shows the final cover fitted.

The top of the robot is painted half white, half black. The vision system determines the robot's orientation by measuring the gradient of the straight line where black meets white. This provides a good estimate of vehicle orientation.

One issue raised at the start of the Robot Sheepdog project was the effect of the robot's appearance on its ability to control the flock. Some work exploring this is described in [Henderson, 1997].

3.4.6 Control design

Rover's control system is in two main parts. A high-speed, low-level classical controller implemented onboard the vehicle regulates the wheel speeds. A slower 'strategic' controller offboard the vehicle guides the vehicle around the arena. The controllers are described here and their efficiacy demonstrated in Section 3.6.

3.4.7 Onboard controller

The onboard controller runs on the vehicle's TMS370 microcontroller. It must minimize the difference between the observed and desired wheel speeds by varying the duty cycle of the PWM speed controllers. Figure 3.4 shows how this controller fits into the system.

Output to a wheel's motor controller at each time step (u) is determined by the proportional controller

$$u = K(S_d - S)$$

(where u is the output to the wheel, S_d is the desired speed, S is the actual measured speed, 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 can maintain wheel speeds that closely approximate those desired. Increasing the controller gain leads to jerky movement as the wheel speed oscillates about the desired value, and at extreme levels wheel slip can occur as the acceleration requires more grip on the dusty floor than the solid rubber tyres can provide. Too low a gain and the vehicle responds to wheel speed demands very slowly. The chosen gain is an acceptable compromise between these extremes.

3.4.8 Offboard controller

The offboard strategic controller runs on the Pentium 200 workstation. It must generate desired wheel speeds for the vehicle, given the current positions and orientation of the vehicle and the position (and possibly size) of the flock. Figure 3.4 shows how this controller fits into the system.

The offboard controller runs three algorithms in sequence; the output of each forms the input of the next:

1. Flock-control - generates the next position that the robot should achieve at each time step.

2. Wall avoidance - prevents the robot hitting the perimeter wall.

3. Steering - determines the wheel speeds required to move to the new target position

1. Flock control

The 'flock-control algorithm' 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 $(R, F, G) \rightarrow \vec{r}$.

The design of suitable algorithms was a major part of this project. Chapters 4 and 5 describe the design and performance of two such algorithms.

2. Wall avoidance

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 as indicated in Figure 3.9. The maximum radius for the vehicle is set to *arena_radius - (robot_radius + safety_buffer)*. A safety buffer of 10cm was found to work well. This simple method proved successful both in simulation and in reality.

3. Steering

The steering algorithm takes the modified target vector $\vec{r_1}$ and generates appropriate left and right wheel speed demands to pass to the vehicle, such that the vehicle will move along $\vec{r_1}$.

The error in heading is determined by comparing $\angle r_1$ with the current heading measured by the vision system. The vehicle operates equally well both forwards and backwards, so the heading

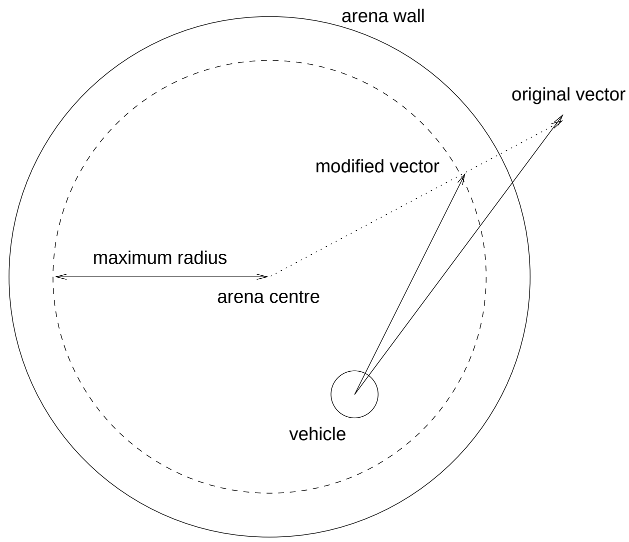


Figure 3.9: Wall avoidance method; any path that would take the robot outside a maximum safe radius is modified to bring it safely inside the arena wall, preventing collision.

error can be at most $\pi/2$rad. The mode (forward or reverse) which requires the least turning is selected here.

The vehicle's desired travel speed (v) and turn-rate (ω) are then determined by the following method, where θ is the error in heading and $|r_1|$ is the distance to the target.

if $\theta > 0.7$rad **then** *TURN:*

rotate on the spot towards the target with constant turn rate. ($v = 0$, $\omega > 0$).

else if $\theta > 0.2$rad **then** *TURN/MOVE:*

turn towards the target at constant turn rate, while moving forwards with speed inversely proportional to θ ($v > 0$, $\omega > 0$,).

else *MOVE:*

move towards the target at speed proportional to $|r_1|$ ($v > 0$, $\omega = 0$).

The turn-rate and vehicle speed are limited to a maximum of 0.3 rad/s and 0.5 m/s respectively. These are well below the vehicle's mechanical limits, but are imposed to keep the vehicle under control. Given a fixed update speed, the error in the controller's estimate of the vehicle's position is proportional to the vehicle's speed, eg. at a realistic update rate of 100ms and a vehicle speed of 2m/s, the position estimate could be out by 0.2m. The time delay is also evident when adjusting the heading of the robot; if the turn speed is too high then it has a tendency to overshoot the ideal heading and perform a weaving path. Keeping travel and turning speeds down enables much more accurate control. Thus the limiting factor in the operating speed of the robot is the update rate of the vision system.

The desired wheel speeds can be calculated from the (modified) travel speed and turn rate by the formula

$$S_{left} = (\frac{v}{t} + r)\omega$$

$$S_{right} = (\frac{v}{t} - r)\omega$$

where $S_{left,right}$ is the desired speed, v is the travel speed, ω is the turn rate and r is the robot radius. The desired wheel speeds are transmitted to the vehicle by the radio modem.

A vision system that measures the position and orientation of the robot in the arena was devised in collaboration with Neil Sumpter and completes the control loop. Effective control is demonstrated in the series of tests presented below, after the vision system is described.

3.5 Vision

The vision system provides the sole input to the off-board controller which guides the robot's movement. The image stream from the video camera over the arena is processed to determine the position and orientation of the robot, and the position and size of the flock. Example images are shown in Figure 3.10. The vision system is designed to be as simple, effective and fast as possible. Its accuracy could perhaps be improved, though probably at the expense of speed, and more sophisticated techniques could have been used. This was not considered necessary as the novelty of these experiments is in the animal/robot interaction and not in the image processing. A description of the system and techniques used is provided here for completeness. Neil Sumpter has produced novel and interesting machine-vision research in a separate stream of the project; see [Sumpter et al., 1997, Sumpter and Bulpitt, 1998, Sumpter et al., 1998].

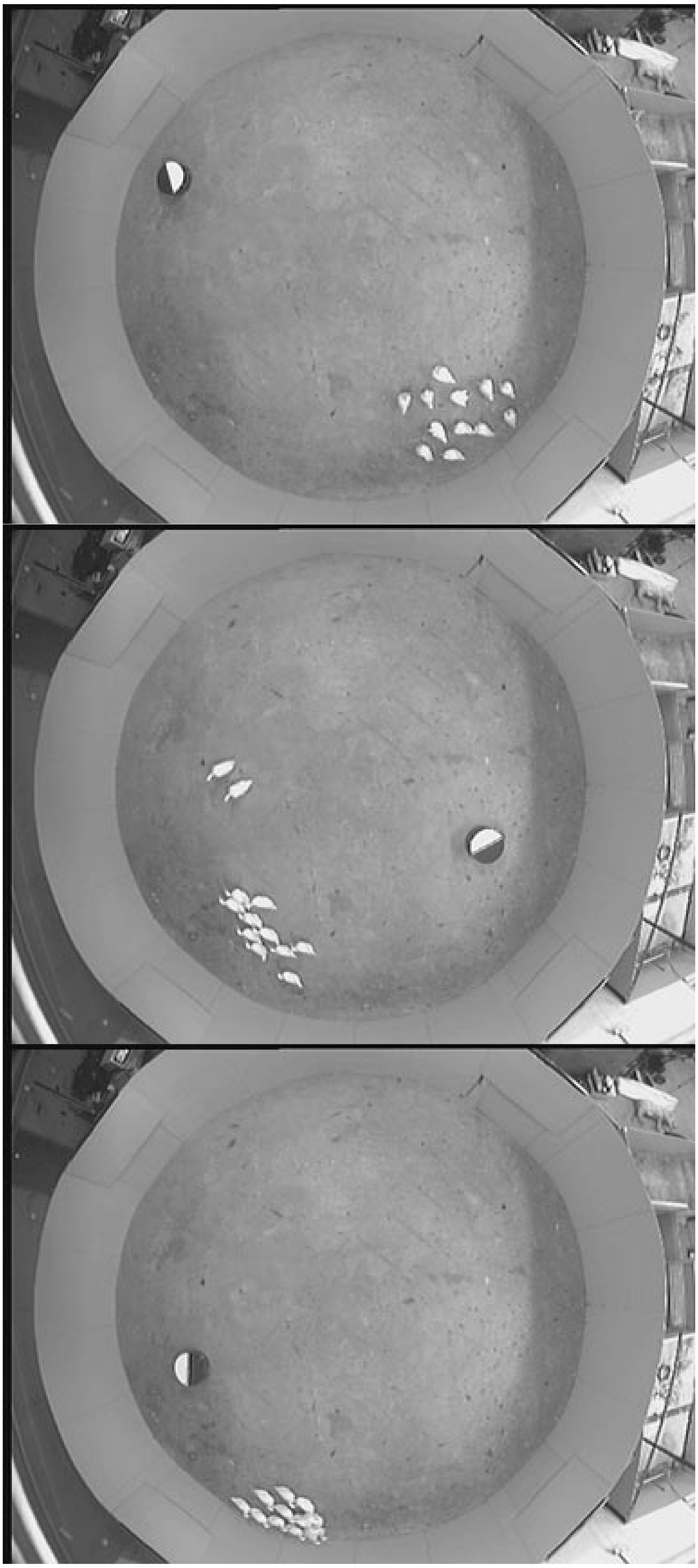


Figure 3.10: Example images from the overhead camera input to the vision system.

3.5.1 Requirements

The vision system is required to locate the position and heading of the robot and flock within the arena, feeding this back to the controller at as high a frequency as possible. At the start of development, the goal update frequency was chosen as 10Hz. It was designed and implemented in collaboration with Neil Sumpter. The subsystems below marked (NS) were largely Neil's work and are described here only briefly.

3.5.2 Image-plane to world-plane mapping

(NS) The video camera uses a wide-angle (3.2mm) lens, and could be mounted only approximately over the centre of the arena. Thus distances measured on the image will not scale directly to real-world distances, but will be increasingly distorted towards the edge of the image. A standard camera calibration package [Tsai, 1987] is used to map points on the image (measured in pixels) to points in the world (measured in metres). For calibration a grid of points of known location was marked out on the floor of the arena. The calibration algorithm produces a mapping function allowing fast conversion of coordinates from image to world planes.

3.5.3 Robot tracker

(NS) The position of the robot is found by identifying the white semi-circle on its lid. For a description of this algorithm see Neil's forthcoming thesis [Sumpter, 1999].

3.5.4 Flock tracker

An ideal system would track the positions of individual ducks. Following discussions with colleagues expert in machine vision it was concluded that there were no reliable, fast methods available to achieve this, certainly given the modest computing resources available. However, it seemed likely that the whole flock could be tracked as one object, with some measurement of its size and shape. Such 'blob detectors' are common in machine vision, and are implemented via textbook techniques such as background subtraction and thresholding (see, for example, [Boyle et al., 1993]). Flock position would be defined as the centre of area of the detected flock 'blob'.

This gave an interesting constraint to the rest of the system; it would have to work without knowing the positions of individual birds, but only with a centre position, size and shape.

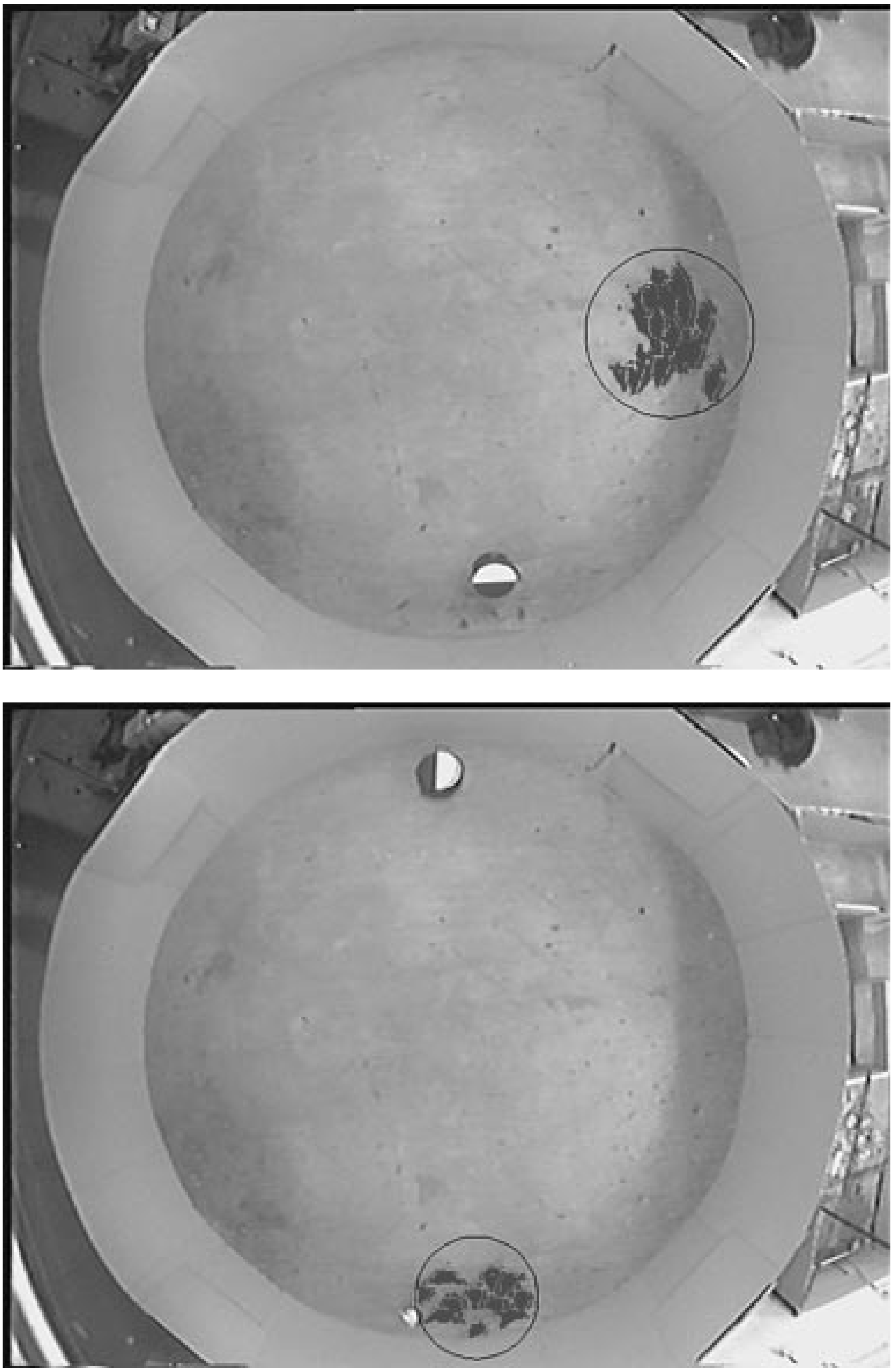


Figure 3.11: The tracker correctly locates the flock

Neil Sumpter developed a tracker which found the flock shape using a combination of image subtraction and hysteresis region growing [Sumpter et al., 1997]. This worked well on test sequences, but unfortunately proved to be slow and unreliable when integrated with the rest of the robot system. A good feature of this tracker was that if it lost the flock it could be manually reset by the operator using the mouse pointer on a window showing the arena image. Adding this user override improved its effective reliability, but its speed could not be easily improved. About half of the processing time was spent calculating the flock shape, so it was decided to abandon the flock shape information, and just determine the position and size of the flock.

The author then developed a second much simpler tracker which was much faster and proved to be reliable *enough* to perform the experiments in this thesis. However it remained the weakest link in the system during these experiments, spoiling perhaps 15% of trials. Further improvements to the tracker were made after the experiments were finished. These solved the main reliability problem (described below) and greatly increased the speed. The final version of the tracker is very effective and is described in Appendix A.

3.5.5 How the flock tracker works

The flock tracker must locate the centre and radius of the flock from the video stream.

Stage 0: initialization

Before a trial starts an image of the empty arena is stored for use as a background reference.

The flock position is initialized by the user indicating where on the image to start tracking. The flock radius is initialized to some small value (say 20 pixels) which will be the minimum allowable flock radius. The maximum flock radius is that of the arena (around 220 pixels). The tracker actually processes a square region around the flock centre, with size (2 x radius, 2 x radius), rather than the implied circle to save a little processing time. All the following processing steps act *inside this region only* and not on the whole image.

Stage 1: flock-pixel spotting

The first stage in the processing removes the background; that part of the image which doesn't change between frames. When the ducks are introduced into the arena, those parts of the incoming image containing a duck (or robot or other new object) usually have different grey-levels than the corresponding part of the background image.

Each incoming pixel in the flock region is compared to the corresponding pixel in the background image. If the difference between the incoming pixel and the background pixel is more than a preset threshold it is considered an 'interesting' pixel. If the interesting pixel is not part of the robot (ie. it is above a threshold distance from the robot position as determined by the robot tracker) it is considered part of the flock. An array is built up of all these flock-pixels.

Stage 2: flock position

The flock centre is recalculated as the mean position of all the flock-pixels. The new flock radius is determined by finding the most distant flock pixel from the new centre, but the change is smoothed by a weighted sum of the newly calculated extreme and the current radius:

$$r_1 = M * r_0 + N * d$$

where $M + N = 1$, r_0 is the current radius, r_1 the new radius and d is the most distant flock pixel. Setting $M = 0.6$ and $N = 0.4$ was found to nicely smooth the changing flock radius.

Iteration

The processing stages 1 & 2 above will not locate the flock in one pass, but must iterate several times to find the best fitting circle around the flock. Each iteration produces a slightly better fit than the last, until no more improvement can be made. However, one iteration is performed very quickly. This is ideal for for processing an incoming video stream.

At high update rates, each frame of video is very similar to the previous one, so the flock is in a similar state. The tracker exploits the continuity of the incoming images: if the tracker can improve its fit faster than the flock can change, then we can track the flock. This sets a minimum frame rate that the tracker must achieve, based on the speed of the objects being tracked. The original goal for this system of 10Hz was sufficient to track the duck flocks used in the RSP experiments. It also successfully tracked any moving object in contrast to the background and larger than a few centimetres; Figure 3.11 shows the tracker correctly locating the flock.

3.5.6 Problem

This tracker suffered a problem which reduced its reliability. The problem was peculiar to the application: the ducks continually created new high-contrast blobs of urine and faeces which the tracker would interpret as duck-pixels. Appendix A illustrates the problem and presents a solution which is found to vastly improve the tracker's reliability. As a happy side-effect the speed is also improved. The experiments performed for Chapters 4 and 5 would have been easier to perform if this solution had been found earlier.

3.6 Robot performance

With the arena, vehicle and vision system constructed, the robot is complete and can move around under its own guidance. The following tests were performed to verify that the robot was indeed under control and could repeat the same movements with reasonable accuracy.

The results presented here are those obtained after a few rounds of informal trials, adjusting controller gains to give an acceptable performance. The goal was to produce a *good-enough* robot. Plots and descriptions are given as a rough indication of the system's performance, rather than as a comprehensive engineering analysis.

3.6.1 Test 1A: goal seek

This test demonstrates that the system can move Rover to a fixed, predetermined goal position. The goal is set at (0,-3) where all positions are measured in metres with the origin at the centre of the arena. The vehicle is placed in the arena at (0,2.5), with its wheels aligned towards the goal (recall that the vehicle has no preferred front, but works equally well in both directions). The trial begins when the controller is started and ends when the goal is reached as defined below. The position of the vehicle in the arena and its distance from the goal are recorded over the length of the trial. The trial is repeated three times to test that the robot's behaviour is consistent over time.

Controller

The only part of the robot control system left undescribed is the highest-level 'strategic' controller (the box labelled 'Herding strategy' in the control schematic Figure 3.4). This controller must take the current vehicle position, decide where the vehicle should be at the next time step and generate a position error, in the from of a vector from the current to desired next positions. In this test, the desired next position is always the predefined goal position, so a controller was constructed that generates simply the vector from the current position to the goal. The identical controller is used in all these tests.

The vehicle is very unlikely to reach the goal exactly; rather it will get very close to the goal and continue to make small adjustments to its position indefinitely. This *dithering* could be eliminated by having a cut-off point in the goal-seeking algorithm (eg. if distance-to-goal is < 0.1m then stop). This would impose a limit on the positioning accuracy of the system. For these tests no such threshold was implemented so the amount of dither could be observed, giving a measure of the positioning accuracy. The robot is considered to have achieved the goal and to be dithering when its position averaged over a few seconds is approximately constant.

Results

Figure 3.12 A shows Rover's path across the arena for all three runs. The paths start out very similar for the first metre, but diverge slightly until they differ by maximum ≈ 0.5m at 1.5m from the goal. In all three trials the vehicle reaches the goal, then dithers slightly back and forth until the trial is stopped.

Figure 3.13 A shows Rover's distance from the goal over time for all three runs. In each case the the vehicle accelerates from rest at 0s until ≈ 2.5s, then maintains a constant speed ($\approx 0.15 ms^-1$) until ≈ 30s. As it approaches the goal, the vehicle decelerates until it reaches the goal at ≈ 42s and begins to dither with an error of ≈ 0.1m of the goal. The trial is stopped at 60s.

3.6.2 Test 1B: goal seek + small obstacle

The vehicle is operating on a rough, uneven concrete floor which will become dirty during later animal experiments. This test and test 1C were performed to assess the effect of bumps and/or debris to the vehicle's path. The test is identical to test 1A, except that the experimenter placed a wooden metre-rule of thickness 7mm under the vehicle's right wheel as close as possible to (0,0). As for test 1A the position of the vehicle in the arena and its distance to the goal are recorded.

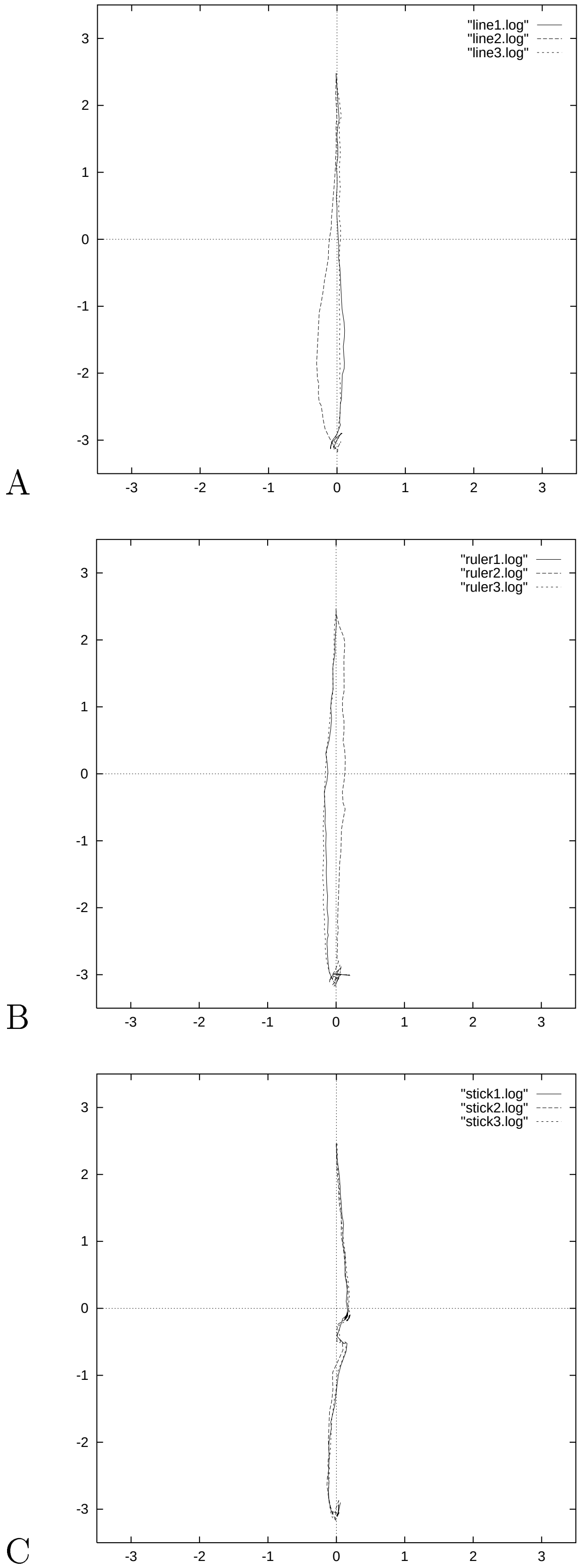


Figure 3.12: Traces of Rover's path as it moves from (0,2.5) to a fixed goal at (0,-3). Right wheel hits an obstacle at approx. (0,0) in B & C. Obstacle size B = 7mm, C = 15mm.

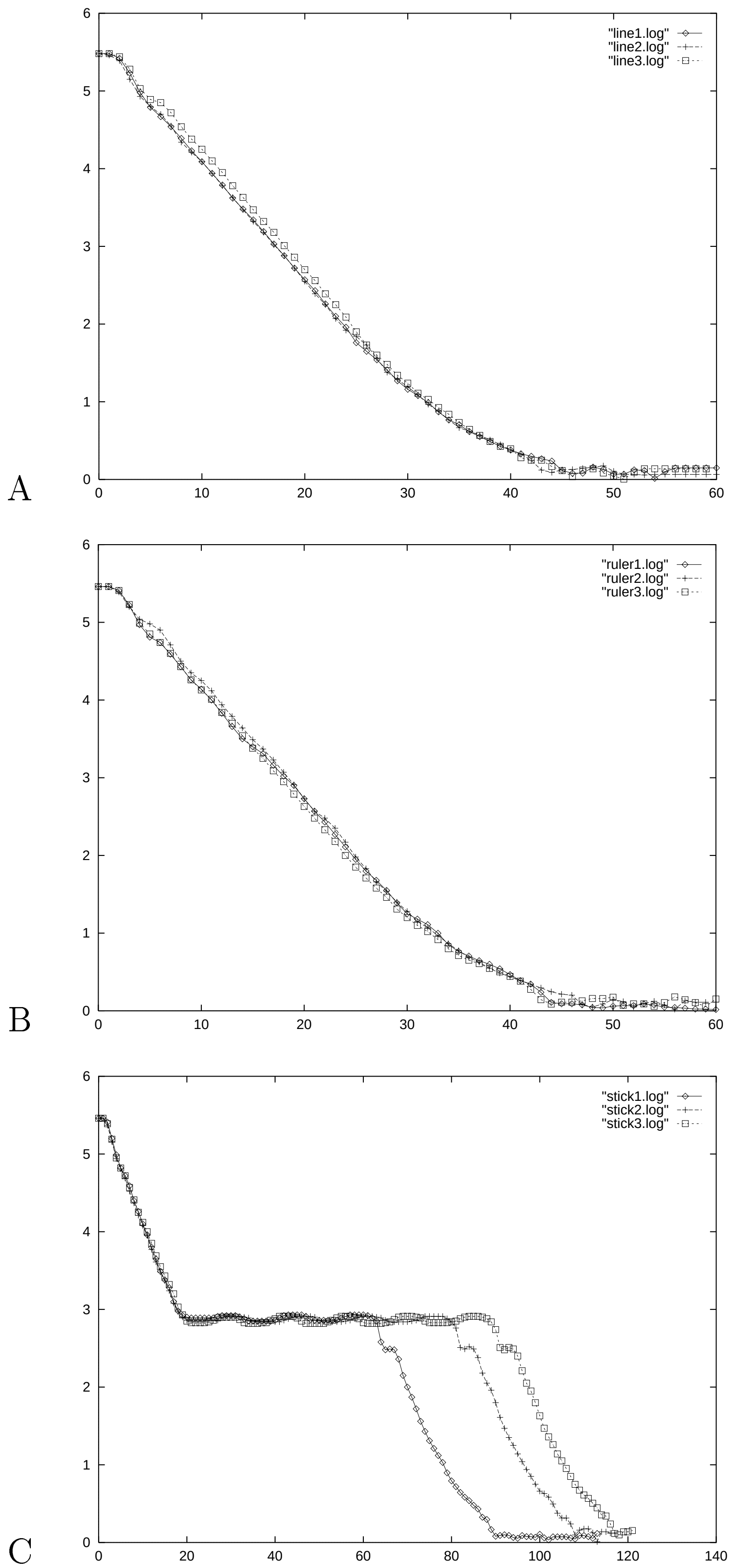


Figure 3.13: Plots of Rover's distance from the fixed goal over time. Right wheel hits an obstacle at approx 20s in B & C. Obstacle size B = 7mm, C = 15mm.

Results

Figures 3.12 B and 3.13 B again show the vehicle's path across the arena and distance to goal over time respectively. The graphs appear very similar to those produced with no obstacle (test 1A) and the vehicle arrives at the target at approximately the same time, indicating that the 7mm obstacle did not significantly effect the vehicle's movement in any of the three trials.

3.6.3 Test 1C: goal seek + large obstacle

This test is the same as test 1B, except that a larger obstacle was used. A 15mm thick wooden bar was placed under the vehicle's right wheel at approximately (0,0).

Results

Figures 3.12 C and 3.13 C show the results for this test. The robot cannot immediately negotiate the 15mm obstacle, but is stopped for 40-60 seconds until the wheel finally clears the bump (Figure 3.13 C). Figure 3.12 shows that the robot is deflected from its path as it hits the obstacle (at $\approx (0,0)$ and 20s) and the right wheel is stopped. As the vehicle begins to point away from the goal, the off-board controller reduces the left wheel speed demand and increases the right speed demand. In the absence of an obstacle this would turn the vehicle to point back at the goal. The on-board controller senses that the right wheel speed is too low (ie. zero) and increases the power applied. The wheel begins to climb the obstacle, but looses grip and spins. The spin sensed as an excessive wheel speed, the power is cut from the wheel and the vehicle falls back to its previous position. This is repeated until the wheel happens to grip the obstacle and clears it. These repeated attempts are visible on the distance-to-goal plot 3.13 C as the oscillations between 20s and 60-95s. As the right wheel clears the obstacle and hits the floor again the power is excessive and the vehicle turns to the left, overshooting the target direction. The off-board controller senses this and turns the robot back towards the goal. This event is visible as the last kink on the distance-to-goal plot. The turns to the right and left are also visible on the position plot, just below (0,0). The vehicle eventually achieves the goal as before.

The failure of the robot to easily negotiate the 15mm obstacle could possibly be rectified by a more sophisticated controller; perhaps an on-board one that identified wheel-slip conditions, or an off-board one that deduced an impact with an obstacle. A more simple solution to this problem could be to use softer, more gripping tyres with the same controllers. In any case, it was shown that the robot dealt well with small (7mm) bumps, and it was considered very unlikely that the vehicle would encounter an obstacle larger than 7mm in any later experiment. In fact, no problems were ever found in practice.

3.6.4 Test 2A: moving target - circle

This test was designed to demonstrate that the vehicle could reliably track a moving goal.

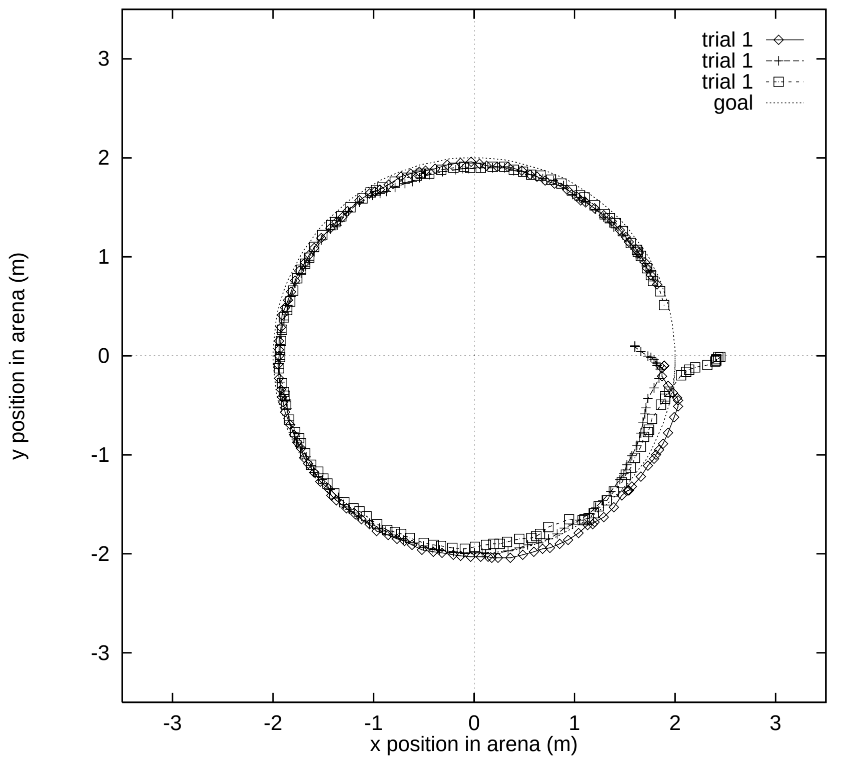
The goal position is initialized to (2,0) and the vehicle placed close by. The test starts when the system is activated, and the goal position is moved in a 2m circle about the arena centre. The goal moves at $0.1ms^{-}1$ and is recalculated every control cycle ($\approx$ 15Hz). The test ends when the goal position completes a whole circle. The robot controller was identical to those used for the previous tests. The test was repeated three times to demonstrate consistent behaviour.

Results

Figure 3.14 (top) shows traces of the vehicle's path in the three tests. From its start position near the goal at (2,0) the vehicle turns and moves towards the goal. As the goal moves clockwise around the circle, the vehicle continues to approach the goal.

As the controller demands a speed proportional to the distance from the goal and the goal is moving, then the vehicle is bound to match the speed of the goal and maintain a constant distance from it (provided the vehicle can vehicle can move faster than the goal). This distance will be determined by the controller gain. If the gain is very high, the distance will be close to zero, and the vehicle will dither about the (moving) goal point. This is undesirable, so the gain is kept low enough to maintain a small distance from the goal, eliminating dither. This can be seen in Figure 3.14 (bottom), where the vehicle maintains roughly the same distance ($\approx$ 0.9m) from the goal for the majority of the trial, following an initial chasing period. Much of the wobble on the this plot is due to estimation error in the vision system; the vehicle's movement was smoother than the plot suggests.

In each trial the vehicle describes a reasonable circle in the arena. A further test was performed to demonstrate more demanding goal-tracking.



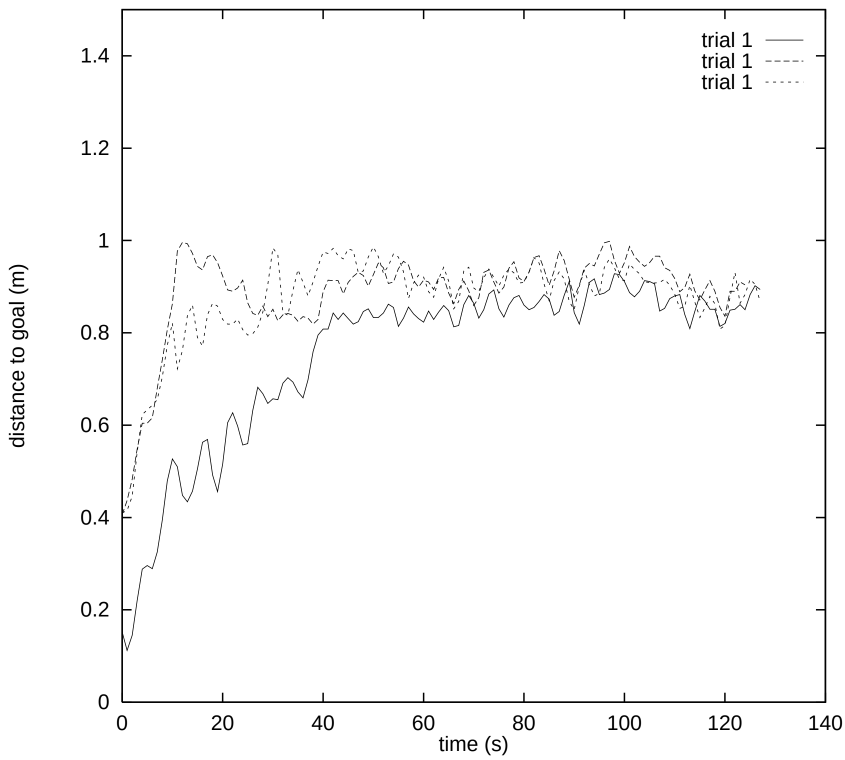


Figure 3.14: Plot of Rover describing a 4m diameter circle.

3.6.5 Test 2B: moving target - square

This test is similar to the previous one, but this time the goal is moved in a 4m side square at $0.1ms^-1$, starting from the top left corner at (-2,2). The vehicle was placed near the start position, the trials began when the system was activated, and ended when the goal had completed a whole square.

Results

Figure 3.15 (top) shows the paths of the vehicle around the arena, starting from the top left. In each case the square was traced reasonably accurately, but the corners were cut off. This is due to the goal-speed matching described above, which produces what is effectively a low-pass filter, smoothing any high-frequency movements made by the goal. The amount of this smoothing is determined by the following distance which is in turn determined by the controller gain as described above. Informal trials showed that increasing the controller gain in an attempt to improve the frequency response introduced undesirable overshoots on the corners.

The distance-to-goal plot Figure 3.15 (bottom) shows the vehicle alternately catching up with the goal point along the edges of the square, then getting further away as the robot slows to manoeuvre round the corners. The average distance-to-goal after the wind-up phase was very similar in each trial and to the circle test 2A, indicating good repeatable performance.

3.6.6 Assessment

The goal-tracking behaviour demonstrated in these tests was considered good enough for the demands of the anticipated experiments. The controller gains were kept unchanged for all subsequent experiments.

3.7 Simulation

This section describes a simulation model that encapsulates the behaviour of the whole robot / duck / arena system. The simulation was required to be similar enough to the real system to allow the design of flock-control algorithms which would transfer to the real world, yet be simple enough to (a) be easily understood and (b) run much quicker than real time. It was hoped that the process of modeling the system, in particular the flock, would inform the design of flock-control methods.

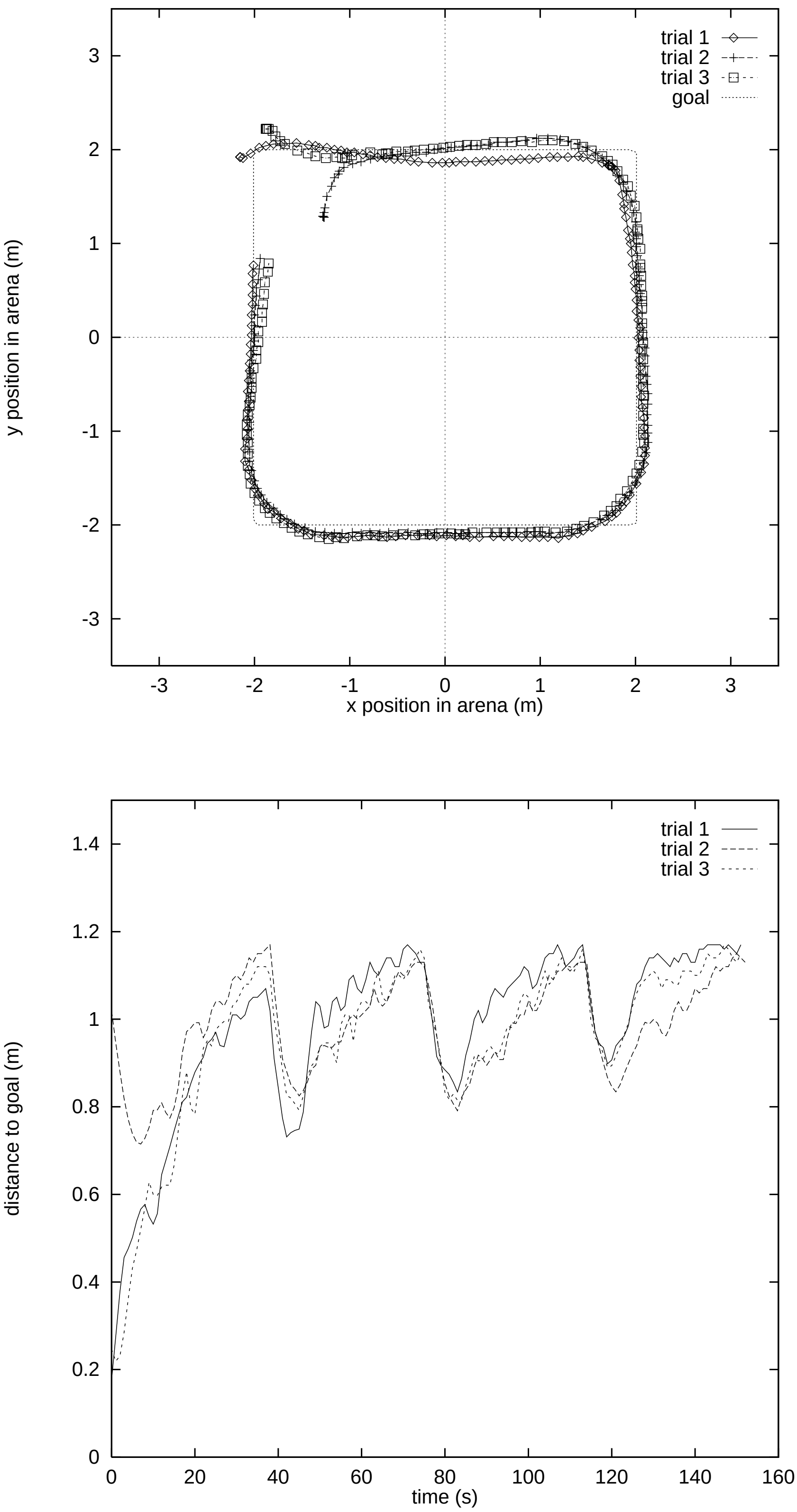


Figure 3.15: Plot of Rover tracing a 4m side square.

The vehicle, vision system and flock are each modeled as simply as possible and incorporated into a single program called 'DuckSim'. The structure and data flow of DuckSim is shown in Figure 3.16; a simplified version of the real robot schematic Figure 3.4.

A clock runs in the world and at each time step the robot and each simulated duck calculates its new position according to the algorithms described below. The simulator increments the clock and moves the ducklets to their new positions. A graphic display shows the state of the world at each step. Each component is described below.

3.7.1 Robot model

The movement of the robot is simulated by a simple geometrical approximation. The steering algorithm described above generates a travel speed and turn rate from which the real robot's wheel speeds are calculated. For efficiency the simulation uses the speed S and turn rate ω directly, calculating the displacement $\left(\frac{dx}{dt}, \frac{dy}{dt}\right)$ and rotation $\frac{d\theta}{dt}$ of the robot in each timestep by the formula:

$$\left(\frac{dx}{dt}, \frac{dy}{dt}, \frac{d\theta}{dt}\right) = (S\cos\theta, S\sin\theta, \omega)$$

The simulated robot thus moves in straight lines rather than the arcs performed by the real vehicle. However, if the timestep dt is kept small with respect to the vehicle speeds, then the true arcs are closely approximated. In practice the simulation is run at 20Hz ($dt = 0.05$s), similar to the 15Hz rate achieved by the real system. This rate is an order of magnitude or more smaller than the vehicle speeds, ensuring realistic movement.

No movement errors were imposed on top of the geometrical model; the model does not simulate wheel slip, bumps on the floor, etc. The reasons for this are similar to those given for the vision simulation, below.

3.7.2 Vision model

Recall that the real vision system takes as input the images from an overhead camera and calculates the position and orientation of the vehicle, and the position and size of the flock. This is simulated by returning the equivalent information from the model robot and flock with perfect fidelity. The robot position and orientation are copied directly from the robot model. The flock centre is

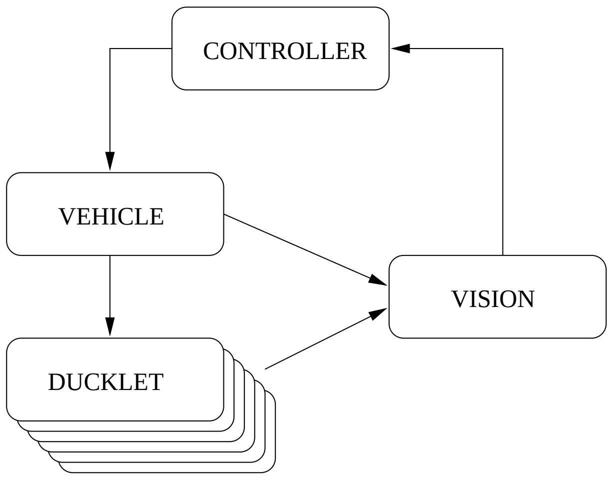


Figure 3.16: Structure of DuckSim, arrows indicate data flow between objects

calculated as the average position of all the ducklets. The flock radius is given as the distance of the furthest ducklet from the flock centre.

Of course, the measurements of the real vision system are subject to error, and some noise could have been imposed on these simulated measurements. However, the noise in the real system may be complex, and not accurately reflected by a typical noise-model such as a Gaussian noise distribution. The nature of the real system noise could conceivably be determined by experiment and analysis, then carefully modeled. This was not done, partly for reasons of time and complexity, but mostly because it was considered that the (lack of) noise from the vision system would be insignificant compared to the other inaccuracies in the simulation. Specifically, the idealized vision system is likely to be more like the real vision system than the ducklets are like the real ducks. The inevitable discrepancies between the model and real flock make high-fidelity simulation of other parts of the system pointless.

3.7.3 Flocking Model

DuckSim models the movement of domestic ducks around a fixed arena in the presence of a threatening stimulus. The flock model developed is related to models in the the literature, described in Chapter 2). It is simplified compared to Reynolds' BOIDS model, but makes fewer assumptions about the processes involved.

Domestic ducks cannot fly, but on the ground they still exhibit strong flocking behaviour in response to a perceived threat. If panicked they may flap their wings, but can not get more than 0.5m or so above the ground in short hops. All the real-world experiments performed were designed to avoid distressing the ducks in this way for welfare reasons, so this mode of behaviour was ignored.

The model attempts to capture a small subset of the behaviour of real ducks. To emphasize this point, the model ducks are called *ducklets*. Of course, many complex mechanisms generate the behaviour of real ducks, but the hypothesis is that this model captures enough of the real animals' behaviour to be a useful design tool. The model is a *generalized* description of flocking behaviour and as such could be applied to any flocking animal in two or three dimensions.

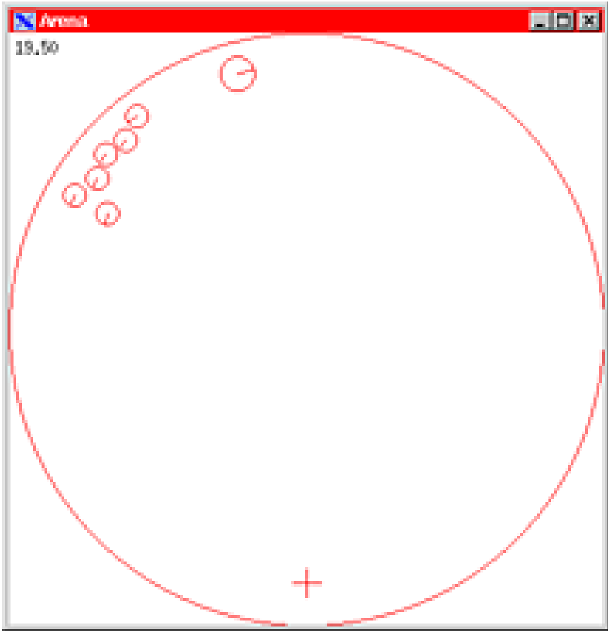
Ducklets are represented by circles centred at the ducklet's position in the arena. All ducklets are the same size and the radius of the circle corresponds to the body size of the ducklet. A ducklet has a front and therefore a heading which is indicated on the display by a 'nose' line. Ducklets, like real ducks, only move forwards, never backwards or sideways. A ducklet is a type of animat (see section 2.3.2). The ducklets must perform an anti-predator flocking behaviour in response to the presence of the simulated robot in the arena. A group of ducklets counts as a flock according to the definition given in section 2.3.1.

Figure 3.17 shows a typical state in the DuckSim arena; six ducklets are aggregated and heading away from the robot.

Mechanism

The movement of the ducklets is determined by a potential field algorithm. This method uses the analogue of a charged particle in an electrical field. The motion of the particle is determined by its own current state modified by the sum of the forces acting it. Several 'forces' act simultaneously between the centre points of the circles:

Given a ducklet's position D, the positions of the N other ducklets $D_{1 \to N}$, the robot's position R and the nearest point on the wall W, the ducklet's potential vector $\vec{d}$ is determined by the function shown in Figure 3.18. 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



Key: ◯ = robot, ◯ = ducklets, $+$ = flock goal

Figure 3.17: Typical DuckSim screenshot.

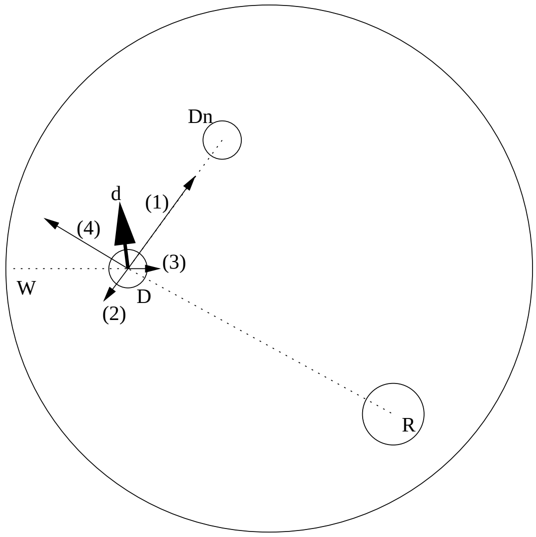
from the robot is proposed to model the aversive response of the ducklets to the robot. All these forces are scaled according to the inverse square of distance (previous models have suggested that this nonlinear response accords with the behaviour in real animals; see section 2.3.2 above).

The resultant of these vector terms is the total force acting on the ducklet. This force produces an acceleration on the ducklet which causes it to move. The acceleration is limited by a simulated momentum, implemented by adding a fixed proportion p of the ducklet's previous movement vector $\vec{d}$ to the forces exerted on the ducklet at its current position $f(x, y)$ to produce its new movement vector $\vec{d_1}$:

$$\vec{d_1} = f(x, y) + \vec{d} * p$$

It is observed that real ducks do not constantly maintain a simple flight distance (see section 2.3.1) from a threat, but that the observed flight distance is actually dynamic, varying for example with the relative speed of the flock and threat. The momentum mechanism is proposed to model this to a simple approximation.

The ducklet's movement is further constrained by imposing a top speed to model the typical



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

(1) (2) (3) (4)

Figure 3.18: Flock model (schematic not drawn to scale). Key: gain parameters $K_{1\to 4}$; repulsion bias parameter L (ensures repulsion > attraction at small distances, preventing collisions); ducklet position D, other ducklet D_n; Robot position R; Nearest point on wall W; algorithm terms $(1 \to 4)$ and resultant velocity $\vec{d}$ (where $\widehat{a}$ is the unit vector of $\vec{a}$).

top speed achievable by the real ducks. The value of this parameter had not been determined for the real ducks when the simulator was created, so a top speed of 0.75 ms^{-1} was chosen as a reasonable estimate. As the robot's speed was to be kept under 0.5ms^{-1} it seemed likely that the flock would not often reach their top speed. Therefore if the chosen value turned out to be a poor estimate, it would not be a significant cause of error. The top speed acts in combination with the momentum to make the observed flight distance dynamic and complex.

Henderson subsequently found that the the flock of ducks used in Section 5.5 (which were of an age and breed similar to those used in all experiments except the pilot trial) achieved an average top speed of 0.74ms^{-1} in treadmill tests [Henderson, 1999, Appendix VII.2], which is in line with the original estimate.

3.7.4 Flight distance and a small arena

In the modelled scenario it is assumed that the robot is always within the flight distance of the ducks, ie. the presence of the robot is always a repulsive stimulus. In her pilot experiments in the arena Henderson found that pairs of ducks would stop feeding and move away from a moving robot-like cylinder as soon as it was presented on the far side of the arena. The ducks never returned to feeding while the robot approached. Henderson concludes that the ducks show a fearful response to the robot at its maximum distance of 7m [Henderson, 1999, Appendix II]. Therefore it seems reasonable for the ducklets to experience repulsion from the robot at any point in the arena.

However, if the arena were arena is much larger (or infinitely large) the ducklets are still repelled from the robot over an arbitraririly large distance. The magnitude of the repulsion quickly becomes very small as the robot-flock distance increases due to the inverse-square scaling, but in principle it is unsatisfactory that the ducklets should be motivated to move away from the robot at *any* distance. To rectify this a simple cut-off could be added to the model; a distance at which the repulsion was forced to zero. This would maximum distance at which the repulsion would apply would correspond to an absolute maximum flight distance.

3.7.5 Justifying the simple scalar motivation model

Clearly the above model does not describe the literal mechanism of a real flock of animals, fish or birds. Rather it is as compact a description of the process of flocking as the author could devise.

As with the model though, the behaviour of an animal at any time must be assumed to be a function of its current state and the effect of its environment. The physiological state of an animal can be characterised as a multidimensional space, where each variable is an orthogonal axis and the origin is at the optimal state. Each axis has a maximum viable value; values for the variable beyond this point are fatal to the animal. Some subset of this physiological space (as it includes the animal's brain) is the *motivational space* which has axes that correspond to motivational cues such as thirst and fear [McFarland and Bosser, 1993]. Of course, the number of dimensions of these spaces is unknown (though likely to be very high) as are the scales of the axes.

In producing a model of animal behaviour, we seek to reduce the dimensions and label the axes of the physiological space to the point where the interaction between the variables becomes comprehensible. This can be done in two ways; (1) by restricting the scope of motivations considered; and (2) collapsing multiple dimensions into a smaller number that approximates the originals.

This model restricts the range of behaviour considered. For example it does not model the reproductive, feeding or social motivations in the ducks. It is assumed that in the close proximity of a threatening stimulus these motivations are overwhelmed by the anti-threat motivation and contribute little to the observed behaviour. This assumption can be shown to be false in certain circumstances, for example there is evidence that ducks maintain some hierarchical position within the flock as they move away from a threat [Henderson, 1999]. By ignoring this, the ducklet model assumes that social hierachies produce no gross changes in the movement of the flock.

There is evidence that real animals show a non-scalar response to the proximity of a threatening stimulus. In fact a sequence of behaviours may be seen in a prey animal as a predator approaches; typically increased vigilence followed by fleeing, protean 'jinking' and finally immobility or even spasmodic fits if the chase is lost [Miller and Cliff, 1994]. The experimental domain in this work is chosen to avoid the high-stress situation that could trigger the later 'emergency' behaviours; a design goal is that the robot must be programmed such that it does not panic the flock. The remaining motivational space of the ducks, however complex in the real animals, is then *approximated* by the single dimensional function where motivation to move away from a threat is proportional to the inverse square of disctance.

Careful consideration of these issues are very important if we are to create an accurate model of duck flocking behaviour. However for the purposes of this work there are three good reasons for using the extremely simple model described above.

1. Simplicity

   The inverse-square term is an elegantly simple way of modelling the aversive stimulus offered by a perceived threat. It has the desired features of being very small at large distances - modelling low perceived threat - and very large at small distances - modelling high perceived threat. Also, as mentioned in the previous chapter, it has been suggested that an apparent inverse-square behaviour could be caused by a linear response to some inverse-square change in sense data [Warburton and Lazarus, 1991].

   Two conceivable simple alternatives are (1) a discreet flight distance outside which an object is not aversive, but inside is aversive to some constant level, and (2) a linear response from some finite maximum aversion at zero distance to zero aversion at some cutoff distance. Both of these require some parameters to be set, the values of which must be estimated in advance and will greatly effect the behaviour of the system. The inverse-square function is preferred due to its lack of parameters.

2. Generality

   A goal of the project was to develop a flock control algorithm that is a as general as possible. Adding features to more closely model *duck* behaviour may well introduce regularities in the simulated flock behaviour that are not present in flocks of other species. Any flock control method designed using the duck model could be exploiting a duck-specific feature and fail to transfer even in principle to other species.

   The chosen approach was to create a method to control the generic flock, then try the method with a particular flock and see if it worked. If this is successful it can be argued that the model captures enough of the important features of the flock to be useful in engineering design.

3. Sufficiency

   The stated aim of the modelling work was to produce a model for use as a tool in the development of a flock controlling robot. It is not claimed that the model extends or improves any model or understanding of flock behaviour. A successful (in the project's limited domain) flock-controlling robot is described in the following chapters, so we can conclude that the flock model was sufficiently good for its purpose.

3.7.6 DuckSim assembled

It should be noted that the the original boids program ran very slowly. Reynolds says that a flock of 80 boids required 95 seconds of Lisp Machine processing time *per frame* [Reynolds, 1987] p32. He does not say how much of this time is spent on generating graphics rather than trajectories, nor exactly how fast the computer was, though a Lisp Machine was a high-performance workstation in 1987 terms (at around 1 MIPS). The Robot Sheepdog system requires DuckSim to run with a flock of 12 ducklets at a minimum of 20 frames per second on a 200MHz Pentium workstation (around 300 MIPS). The model is deliberately minimal and the code has been written to achieve a current update rate of $\approx$180Hz with graphic display and $\approx$4200Hz with graphics switched off.

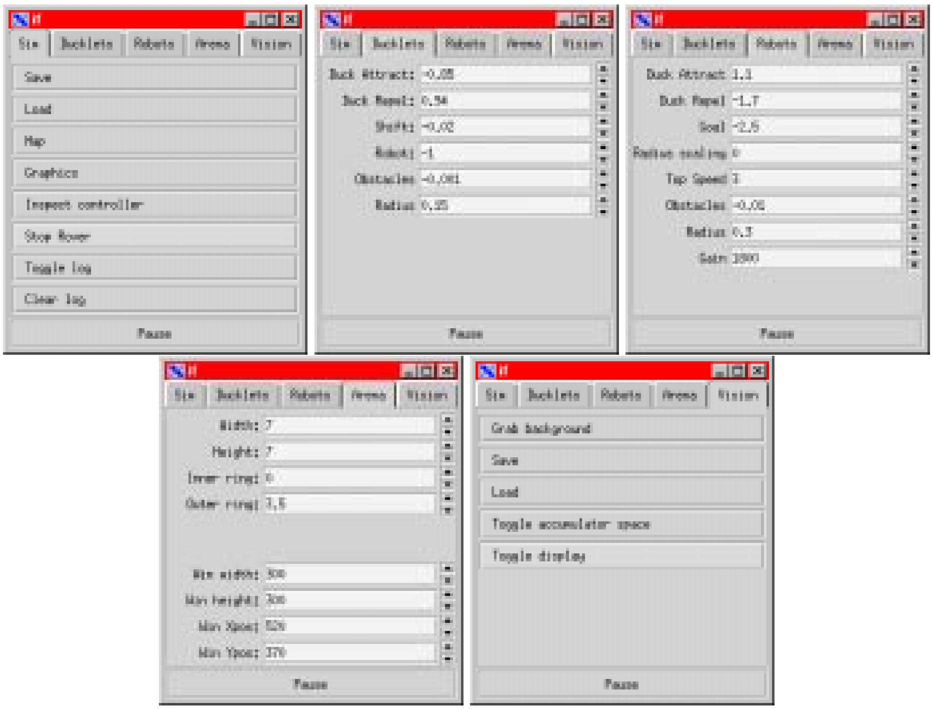


Figure 3.19: DuckSim's TCL/Tk interface, used to control the simulation, adjust system parameters on the fly and to edit the configuration file.

Thus it is possible to run simulated herding trials much faster than real time. There is scope for further code optimization.

DuckSim includes an X11 interface for visualization and adjusting parameters. The DuckSim window gives a view of the arena equivalent to the view from the overhead camera. Ducklets are represented by circles centred at their positions in the arena with a radius equivalent to the size of the real ducks. The simulated vehicle is represented in the by a larger circle with a radius equivalent to the size of the real vehicle's cover. Ducklets can be repositioned with the mouse and the robot can be controlled with a joystick.

A small TCL/Tk program gave a simple interface to the the model's parameters. These could easily be viewed, modified and saved to a file. The TCL program communicated with the main DuckSim application through a network socket, allowing parameters changes to be immediately reflected in the simulation. Experimenting in this way allowed the parameters to be adjusted to give the most realistic-looking flock. Figure 3.19 shows this TCL interface.

To aid visualization of the forces acting on a ducklet, a map can be plotted of the magnitude of force acting in the arena. Figure 3.20 (A) shows a typical DuckSim scenario, with six ducklets

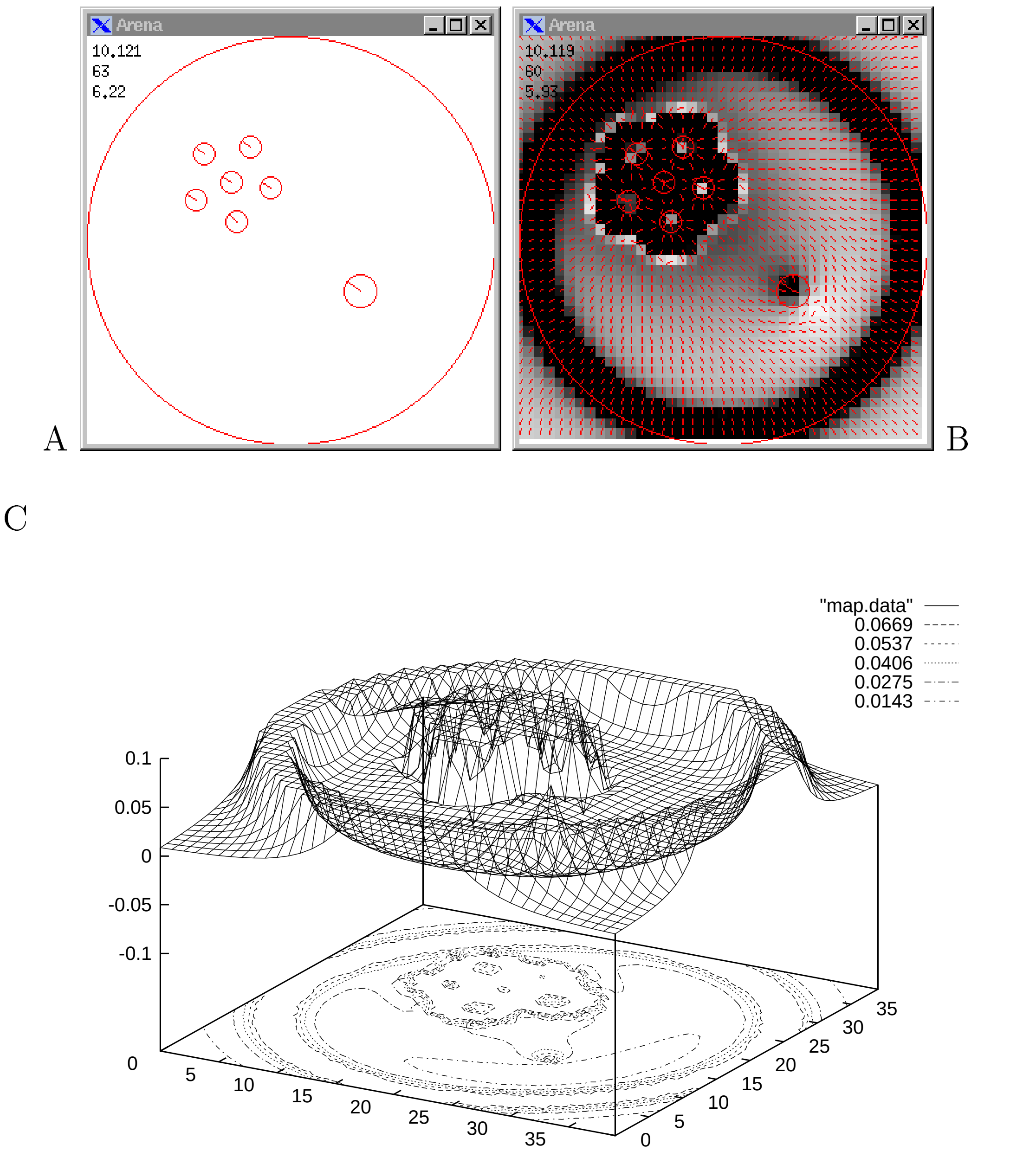


Figure 3.20: Screenshot of a typical DuckSim scenario (A), map of ducklet potential indicating directions (B) and three dimensional representation of the same potentials (C).

moving away from the approaching robot. 3.20 (C) shows a 3-dimensional surface representation of the 'force' that would act on a ducklet at each point on a grid in this scenario. The magnitude of force is shown as height (the heights shown have been cut-off at a threshold for clarity). A ducklet would move as if it were a marble dropped into this *potential landscape*. Potential maps can be generated as the simulator runs and collected into animations for later analysis. Figure 3.20 (B) shows another representation of the same scenario, where the magnitude of force experienced at a sample point is indicated by its shade ($white < grey < black$) and the direction indicated by the small sticks in the centre of each sample. This information was useful for debugging and choosing suitable parameters for the simulation.

3.7.7 Assessment

The intention is that this model captures the behaviour of the *generic* flock; no attempt has been made to carefully match the model's behaviour with that of the real ducks. A quantitative evaluation of the real flock was beyond the scope of this work and was not considered desirable anyway; if the model was too carefully constructed, then any robot controller developed to interact with it could be relying on specific features of *this flock*, and not be a general method. The desired general, robust flock-control methods should be able to cope with the differences between the simulation and reality. This approach to designing 'good' systems with 'bad' simulations is in line with Jacobi's 'radical envelope of noise hypothesis' [Jakobi, 1998].

However, a favourable subjective assessment of the model can be seen in that observers immediately recognize flocking in the ducklets, and judge it as fairly realistic. The attractive features of Reynolds' boids model such as realistic flock splitting have been retained, while the model is considerably simpler and runs much faster.

A technique for splitting the flock up was discovered in DuckSim: maintain the robot at the centre of mass of the flock. Repeating this (approximately by eye) with the robot and real ducks gave qualitatively very similar results.

3.8 Summary

This chapter described the development of Rover, the project's vehicle. Despite its unusual high-speed performance, Rover is simple and of low cost. The vehicle has proved reliable and tolerant of outdoor conditions. The same cannot be said for the vision system, which is the main constraint on the system's overall performance.

The distributed control system is nicely modular, with its layout and communications scheme suggested by observations of the real herding system of shepherd and dog.

The project's simulator 'DuckSim' was introduced and its mechanisms and implementation described.

The last item on the list of required system components was an algorithm to control the robot to drive the ducks. The following two chapters each introduce a candidate algorithm, and describe the results from experiments in simulation and in the real world.

Chapter 4

Flock control 1

The previous chapters have described a robot system that was designed to interact with a flock of ducks. This chapter introduces an novel algorithm that guides the robot to gather the ducks and manoeuvre them to a pre-determined goal position against the arena wall.

Work presented in this chapter has been published as refereed conference papers [Vaughan et al., 1997] [Vaughan et al., 1998c] [Vaughan et al., 1998b].

4.1 Hypotheses

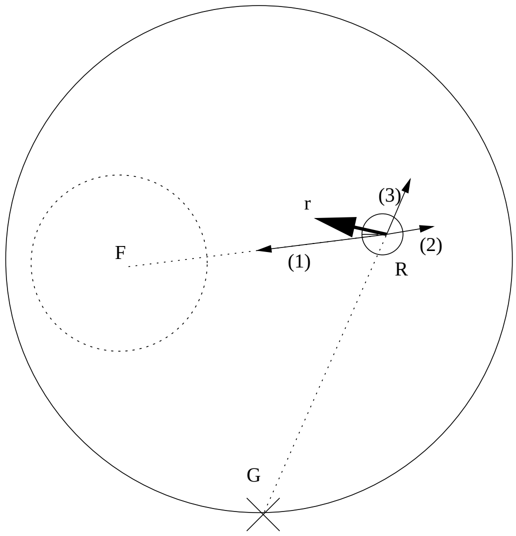
The experiment described in this chapter was designed to test the following trail of hypotheses:

(1) Flock control can be achieved by exploiting the animals' threat avoidance behaviour, whereby the flock moves directly away from a threatening stimulus.

(2) Following on from (1), the appropriate interaction is to position a threatening stimulus behind the flock with respect to the goal. The distance from the stimulus to the flock should be small enough to motivate the animals to move, but large enough that they do not suffer unnecessary stress.

(3) A simulated flock could be used to design and test a robot controller that achieves (2). If the model flock captures the underlying mechanism of flocking, then an algorithm that controls the model's behaviour should control the real animals.

An algorithm is presented that positions the robot vehicle as suggested in (2). A pilot trial tests the algorithm with the model flock described in Chapter 3 and shows that the flock does



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

$$(1) \qquad\qquad (2) \qquad\qquad (3)$$

Figure 4.1: Method 1 (schematic not drawn to scale). Key: gain parameters $K_{1\to 3}$; flock centre F; Robot position R; Goal position G; algorithm terms $(1 \to 3)$ and resultant $\vec{r}$ (where $\widehat{v}$ is the unit vector of $\vec{v}$)

reach the goal position. A similar trial with the same controller running on the physical robot and real ducks shows similarly successful behaviour.

Having demonstrated the system once, a formal experiment was designed to examine its robustness. Metrics for gauging the success of the trials were devised, and a summary of the results of twelve simulated and nine real-world trials are presented.

The flocking model described in Chapter 3 is based around a *potential field* algorithm. Such algorithms are often used in robot navigation. The commonality of these animal and robot behaviour models forms the basis of an effective flock-gathering strategy.

A novel potential field flock-control algorithm was developed by experimenting with the simulator. Potentials were added one by one and parameters adjusted in an iterative process. Therefore there was no definitive pilot trial, but rather many tens of runs until a solution was found. The algorithm described below was the first to successfully control the simulated flock.

4.2 Algorithm

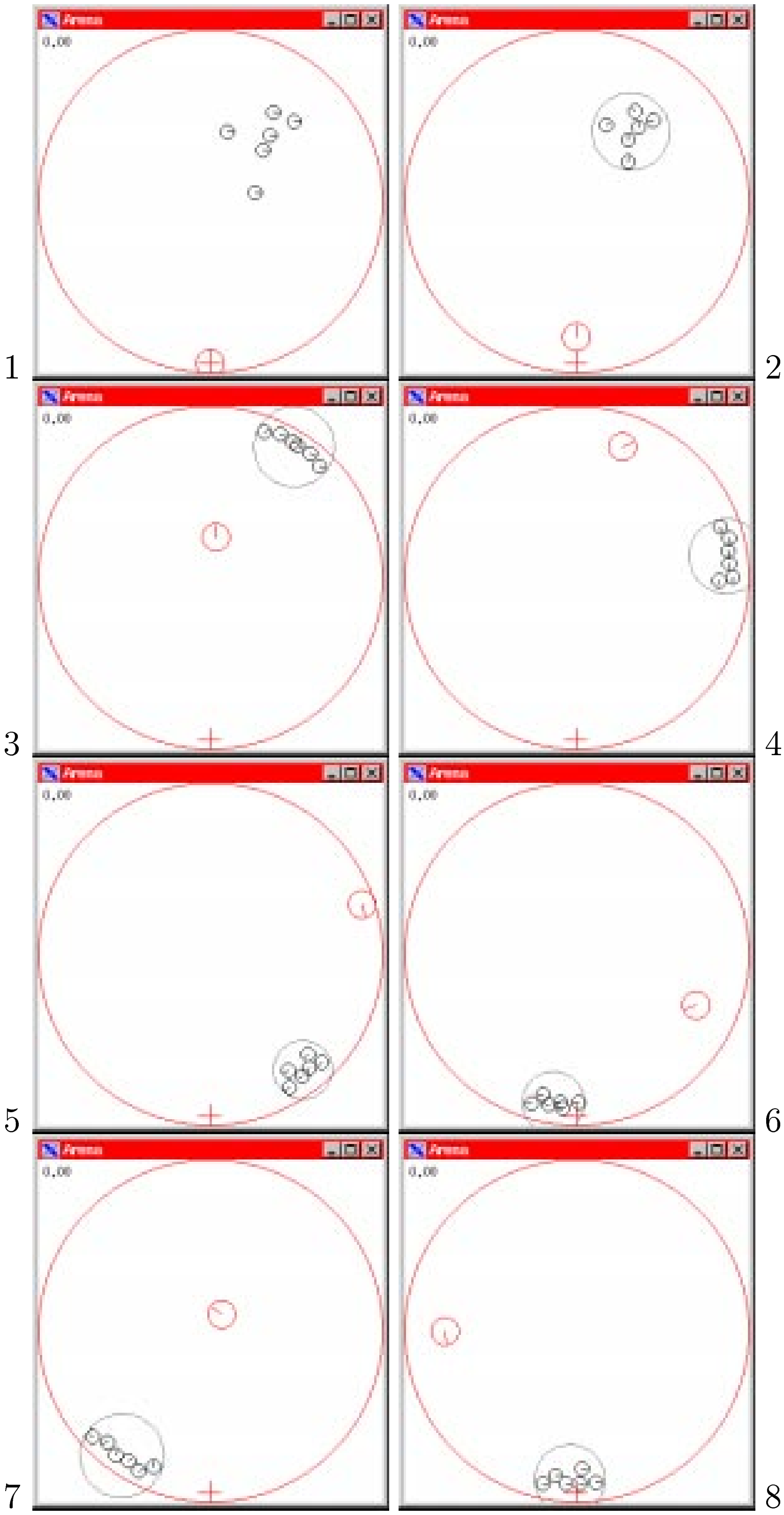
At each time step the flock-control algorithm must generate a movement vector for the robot's low-level controller to execute. In this algorithm the robot's movement vector $\vec{r}$ is given by the function shown in Figure 4.1. The robot is (1) attracted to each ducklet with a magnitude proportional 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 characteristic 'Mexican hat' shaped field, with a circle of zero potential around a positive potential peak at the flock centre. 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 circle around the flock now has a minimum behind the flock with respect to the goal; lifting the 'hat' at the brim. The robot will move around the low-potential orbit to the point of lowest potential behind the flock with respect to the goal. The flock moves away from the robot and towards the goal. As the flock moves towards the goal, the point of low potential follows, pulling the robot with it.

This algorithm will be referred to as Method 1.

4.3 Example simulation trial

A point near the arena boundary is chosen as the flock goal and the robot positioned at this point. Six ducklets are placed in a loose cluster at random in the arena. The simulation starts and the robot moves under control of the Method 1 algorithm. The behaviour of the system is observed until it stabilizes, ie. goes into a repetitive pattern.

Figure 4.2 shows snapshots from the simulator during a single trial, with the snapshots labelled 1 to 8 in time sequence. This example shows that the controller performs the required task with some success. 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



Key: ◯ = robot, ◦ = ducklets, + = flock goal

Figure 4.2: Sequence of DuckSim screenshots showing the simulated Rover fetching the ducklets to the goal.

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.

4.3.1 Conclusions

At this stage the Method 1 algorithm looked promising. Informal experiments with the simulation showed that it works over of range of flock parameters. The oscillation of the flock about the goal is not ideal; it would be preferable for the flock to just stop at the goal. If it is assumed that the ducks are moving in response to a fearful stimulus, then any excess movement implies excess stress on the animals. However, the simplicity of the method was attractive and it seemed directly to reflect the hypotheses suggested above. It was determined that Method 1 would be tested on the real world system.

4.4 Pilot real-world trial

Method 1 was implemented on the real robot, and the simulated trial was reproduced as closely as possible in the real world. Eighteen ducks were raised from hatching in groups of six under controlled conditions; they had the minimum possible contact with humans and no contact with machines before the trials began at five weeks of age. For the week before the trials began they had been introduced into the arena for 30 minutes a day to allow them to become accustomed to the experimental procedures without the robot stimulus. See [Henderson, 1999] for full details of the ducks, handling procedures and the later ethological experiments they performed.

4.4.1 Procedure

A point near the arena boundary is chosen as the flock goal and the robot positioned at this point. Six ducks are introduced into the arena and are free to move. The ducks are allowed to settle for two minutes. The robot is then activated and the trial begins. The positions of the robot and flock centre are recorded until the behaviour stabilizes, ie. goes into a repetitive pattern.

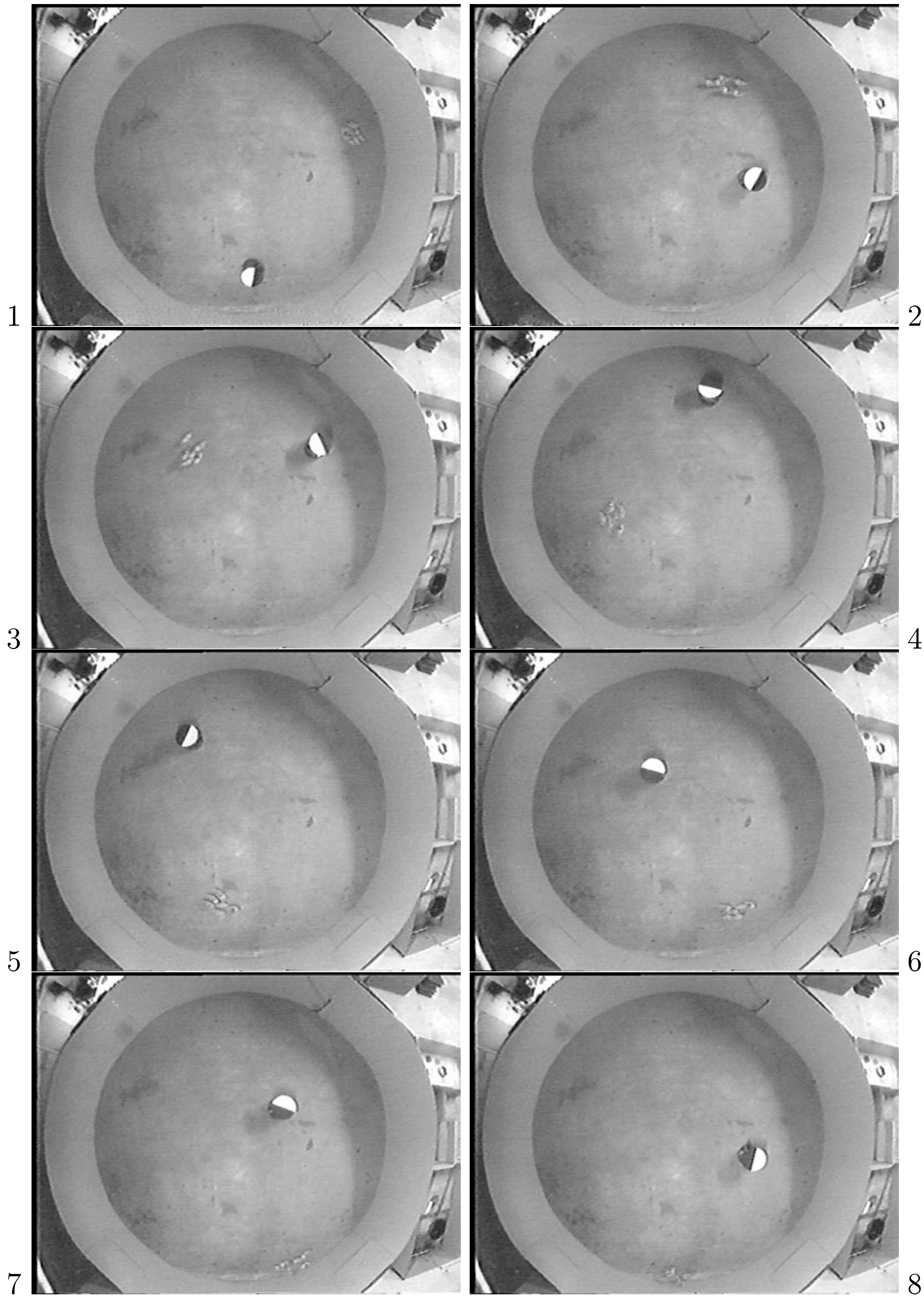


Figure 4.3: Sequence of images from the overhead camera during a Method 1 experiment, showing successful behaviour. The goal position is at the bottom of the picture. The brown ducks are hard to see, so subsequent trials were performed with a white breed.

4.4.2 Results

Figure 4.3 shows images from the overhead camera during the pilot trial. The ducks used were a brown breed with little contrast in shade form the arena floor, so are difficult to see in the pictures. This also caused problems for the vision system as described below.

As the trial begins, the flock is aggregated on the far right side of the arena relative to the goal and the robot (1). The robot moves towards the flock and they move away towards the top of the arena (2). As the ducks approach the top left side of the arena they move away from the arena wall, still away from the robot but not in the optimal direction (3,4). This is likely to be because the ducks could smell or otherwise sense the author operating the workstation just outside the arena wall in this corner. This would not happen in the simulation, where such extraneous effects are not modeled. Nevertheless, the robot moves to the left of the arena in order to position itself behind the flock with respect to the goal (5). The ducks overshoot the goal, so the robot moves across the arena to the right (6,7), again pushing the ducks to the left towards the goal. As the robot slows to turn back across the arena, the ducks stop very close to the goal (8) and the trial is manually halted.

A plot of the flock centre and robot positions over the length of the trial is shown in Figure 4.4. The poor performance of the vision system is evident in the 'glitches' in the flock path plot. The large glitch at roughly (-0.4, -2.5) (as before, positions are given in metres from origin at arena centre) was caused by the tracker losing the flock until it was manually reset. The tracker was subsequently improved for further trials (then drastically improved after the trials were complete as described in Appendix A).

It can be seen from the plot that the flock ends the trial close to the goal position; the robot succeeded in gathering the flock. This first result was particularly interesting because the flock did not behave in exactly the same way to the model used to design the algorithm, but showed additional behaviour in reacting to the presence of a human outside the arena. The algorithm worked by effectively exploiting the threat avoidance behaviour and proved to be robust enough to cope with additional behaviour which was not considered in its design. This robustness is a necessary quality of an Animal-Interactive Robotics system, because the complexity of the real

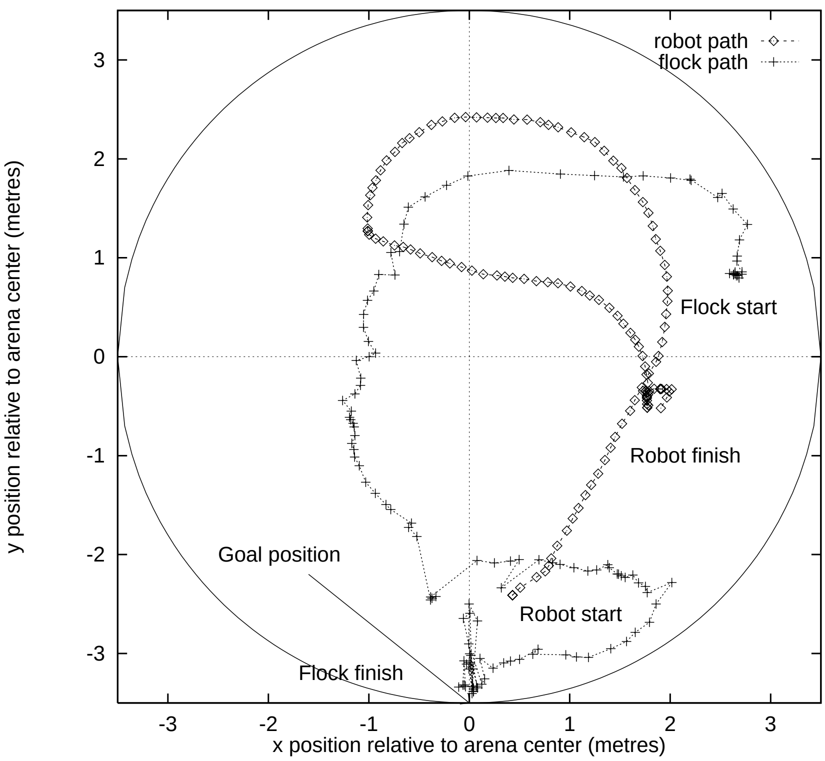


Figure 4.4: Plots of the robot and flock paths through the arena in the Method 1 pilot trial.

animal system cannot be fully modeled.

Figure 4.5 shows the distance from the flock centre to the goal point over the length of the trial. An initial chase phase can be seen as the flock gets further from the goal, up to around 10s, then the flock is driven steadily towards the goal until around 33s. The overshoot is then clearly indicated as the flock-to-goal distance again increases until the robot brings the flock back to the goal at around 53s. The sharp peaks in the plot are again artifacts of the early vision system.

4.4.3 Conclusions

Method 1 seems qualitatively to work in simulation and at least once in the real world. The rest of this chapter presents the results from controlled, repeated trials of the algorithm, so that its performance and robustness can be compared in the simulation and in the real world.

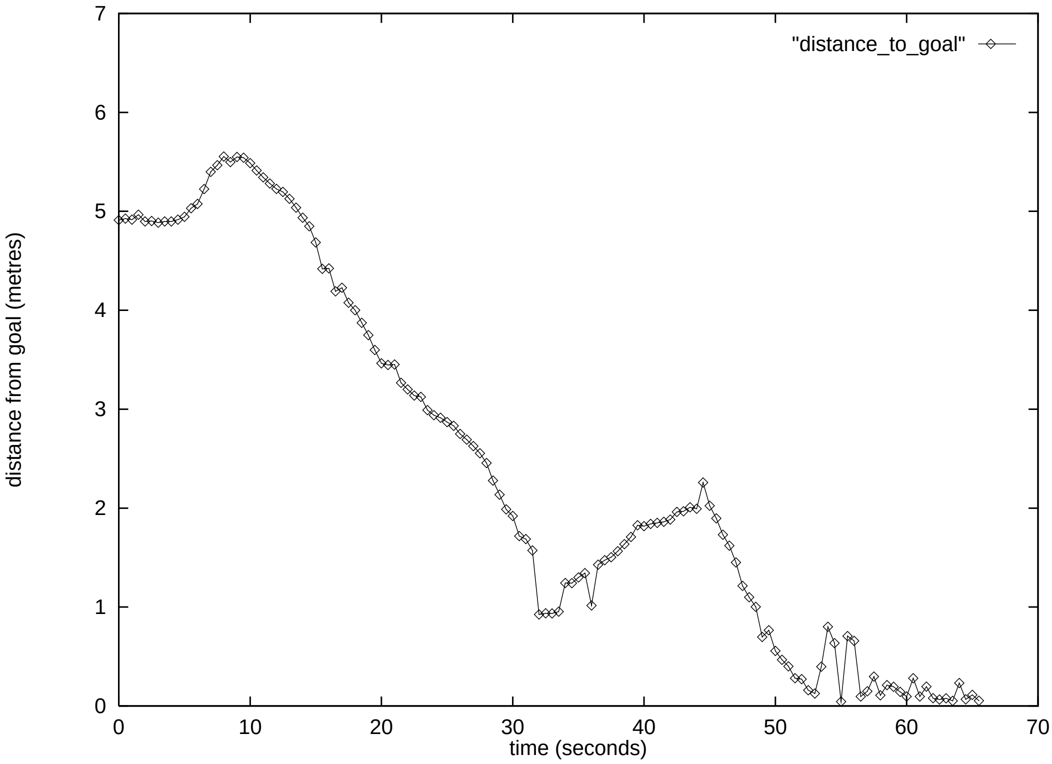


Figure 4.5: Plot of the flock distance-to-goal over the length of the Method 1 pilot trial.

4.5 Simulation trials

4.5.1 Procedure

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 centre are recorded for the next 3 minutes, as the robot attempts to manoeuvre the flock to the goal. Unlike the pilot trial above, the robot is not manually halted if/when the ducks reach the goal, allowing continued observation of the systems' behaviour.

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

4.5.2 Quantifying the results

The flock distance-to-goal plot Figure 4.5 gives a useful impression of the system's effectiveness, but in order to compare trials some metric of success is required. A successful run should bring the ducks as close to the goal as possible as quickly as possible. These criteria are captured in a

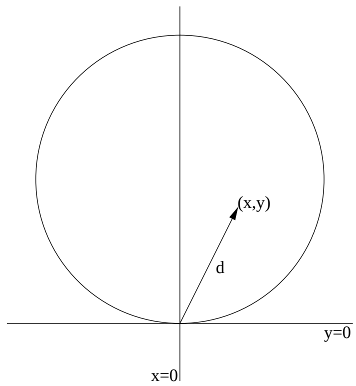


Figure 4.6: An example point (x, y) inside the circle and its distance r from the origin. The null success score r_{avg} is the average distance of all the possible points (x, y).

success score defined as the average distance of the flock to the goal point over the length of the trial.

It will be useful to determine the null success score, i.e. the average success score achieved by ducklets positioned randomly in the arena. If the experimental result show a significant difference to this, it can be concluded that the robot is having some effect on the position of the ducklets.

Calculating the average distance to goal of a large number of random points is equivalent to finding the average distance to goal for *every point* in the arena. This is a special case of the complex general problem of finding the average distance from an arbitrary point to all other points in a defined space. In this constrained case, finding the average distance r_{avg} of all points in a circle to the origin, where the origin is located on the boundary of the circle (Figure 4.6), can be expressed by the integral:

$$r_{avg} = \frac{1}{\pi r^2} \int_0^r \int_0^{-r+\sqrt{r^2-x^2}} (x^2 + y^2)^{\frac{1}{2}} \, dy \, dx$$

[Kalyani Sukhatme, NASA Jet Propulsion Laboratory, personal communication 1999]

We choose to find an approximate solution numerically by experiment. Generating 1 million random points inside a 3.5m radius circle, the average distance to a single point on the edge is found to be 3.962m, or 1.132 times the arena radius. Repeating the experiment with various circle sizes confirms that the result is always this constant multiple of the radius.

Any group of real ducks will not be randomly distributed around the arena. Rather, their position is a consequence of their behaviour which is a function of their internal state and the influence of stimuli in the environment. These factors may be obscure or invisible to the experimenter, but they are not randomized between trials. For example, the ducks may find a spot in the arena which has some scent deposit from previous trials, or has the most comfortable temperature. These effects are difficult to measure and/or control for. Moreover it is assumed that during the trials the effect of these stimuli are insignificant compared to the duck-robot interaction. Therefore the experimental results will be compared with the idealised random distribution described above. In addition Section 6.4 describes a simple control experiment which measures the 'success' of three real-duck trials in the absence of the robot.

A further metric was devised in order to get a measure of the system's **efficiency**. This is defined as the distance travelled by the flock plus the distance travelled by the robot over the length of the trial. The distance traveled by the robot is assumed to be roughly proportional to the battery power expended over the trial, and hence an appropriate efficiency measure. The distance traveled by the flock is assumed to be related to the threatening stimulation they perceive from the robot, and to the energy they expend. In terms of the ducks' welfare, it is assumed that less threatening stimulation and corresponding energy expenditure is better.

By these metrics, the perfect (impossible) trial would score 0 success, 0 efficiency. The worst possible success score is 7m (the size of the arena), and the worst possible efficiency score would be the sum of the distances traveled by the robot and ducks running at top speed for the length of the trial, estimated to be around 1000m.

Determining an a priori null efficiency value is problematic, because the efficiency metric is intended to assess the work done by the robot in addition to the stress applied to the flock. A null efficiency score would be a prediction or measurement of *independent* flock and robot behaviour, but this cannot be found in the model as it stands. The robot only moves in response to the proximity of ducks or ducklets. In their absence its 'efficiency' is essentially meaningless. Flock movement around the arena only occurs in response to the proximity of the robot. It could be proposed to add a random movement of the flock corresponding to the random positions assumed above, but then the average efficiency will depend entirely on the upper bounds chosen for duck speed. In this situation the meaningfulness of this metric is not convincing. Such additions and assumptions are undesirable as we proposed only to model the interaction between flock and robot and not the flocks' independent behaviour.

Comparison can instead be made with the flock movement distances recorded in trials with real ducks in the absence of the robot (Section 6.4). In those trials, the ducks moved an average of 17.7m. When comparing results obtained in robot/duck interaction trials with the no-robot controls, only the flock distance travelled is used; the robot distance is discarded.

An alternative approach to obtaining null success and efficiency scores would be to measure the behaviour of the system with the robot exhibiting some kind of non-interactive behaviour. For example it could follow a fixed route or make random turns when encountering a boundary. This would provide information about the flocks' responses to the robot which are hidden in the experiments that follow because of the tight feedback between flock and robot behaviour. However, the experimental work was completed before this was considered.

Designing metrics to describe behaviour is a thorny task for ethologists and roboticists alike. Martin and Bateson provide a standard reference for ethologists [Martin and Bateson, 1993b] but the robot literature is lacking in such resources.

4.5.3 Results

Results are presented for each trial in Figures 4.7 to 4.18. A plot of the paths of the robot and flock centre through the arena is given (top). The path plots are labelled with the robot start position **RS**, robot finish position **RF**, flock start **FS**, and flock finish **FF**. The plots have been rotated so that the goal position is always at the bottom of the arena at (0,-3.5).

Also given are plots of the flock distance-to-goal (left) and the distance traveled by the robot and the flock (right). These data are used to calculate final success and efficiency scores for each trial. These scores are presented in Table 4.1 along with their averages and variances.

For comparison with later trials, the average success score over 12 trials for Method 1 in simulation was 2.19m. The average efficiency score over 12 trials was 31.86m, of which 12.95m was flock movement.

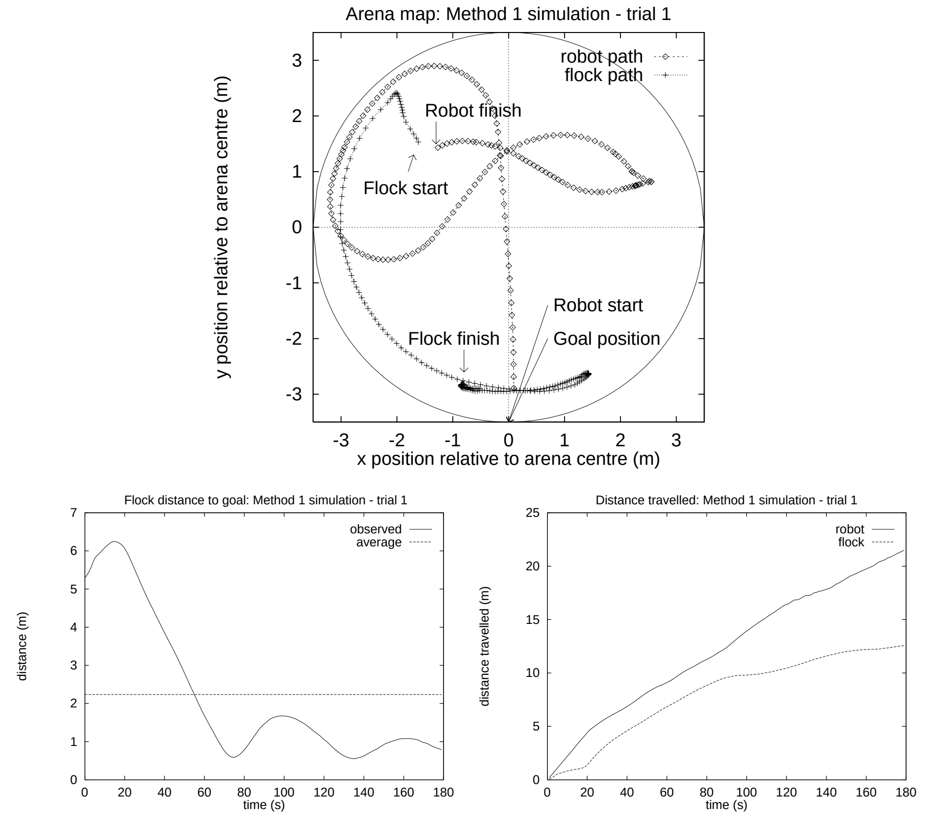


Figure 4.7: Method 1 simulation results - trial 1.

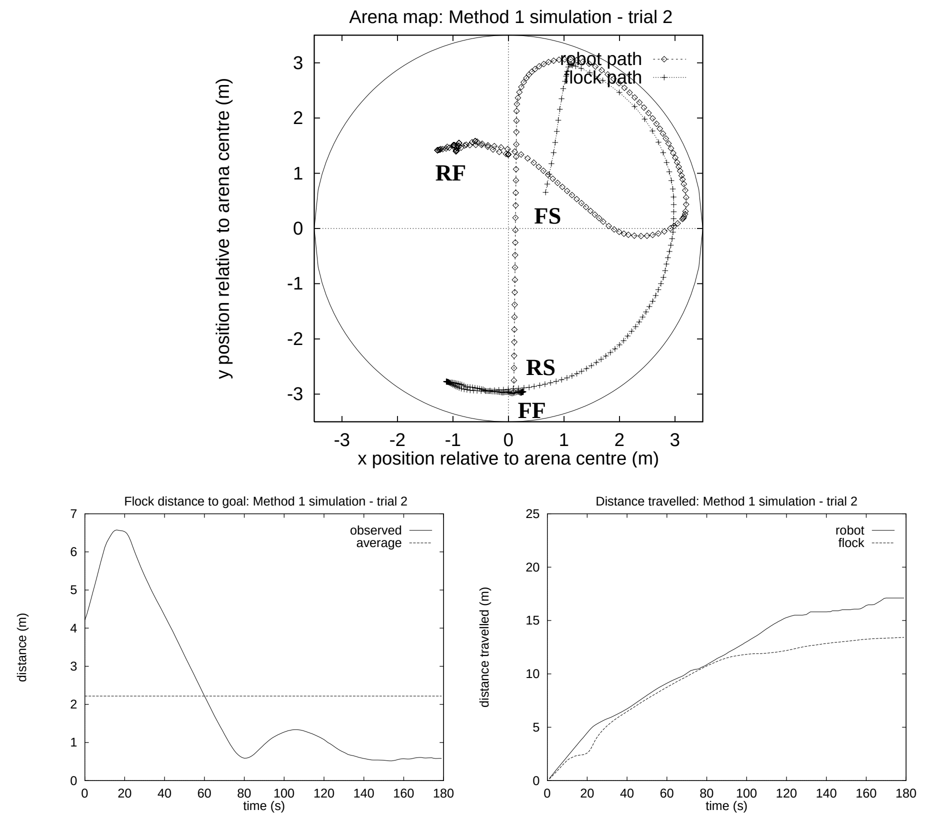


Figure 4.8: Method 1 simulation results - trial 2.

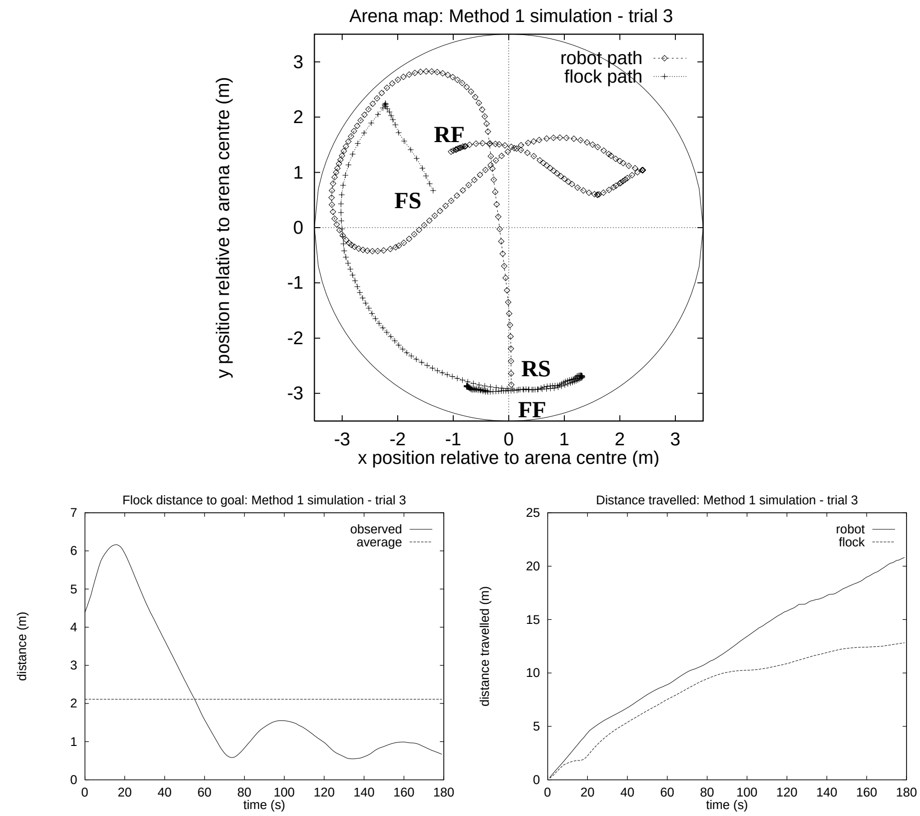


Figure 4.9: Method 1 simulation results - trial 3.

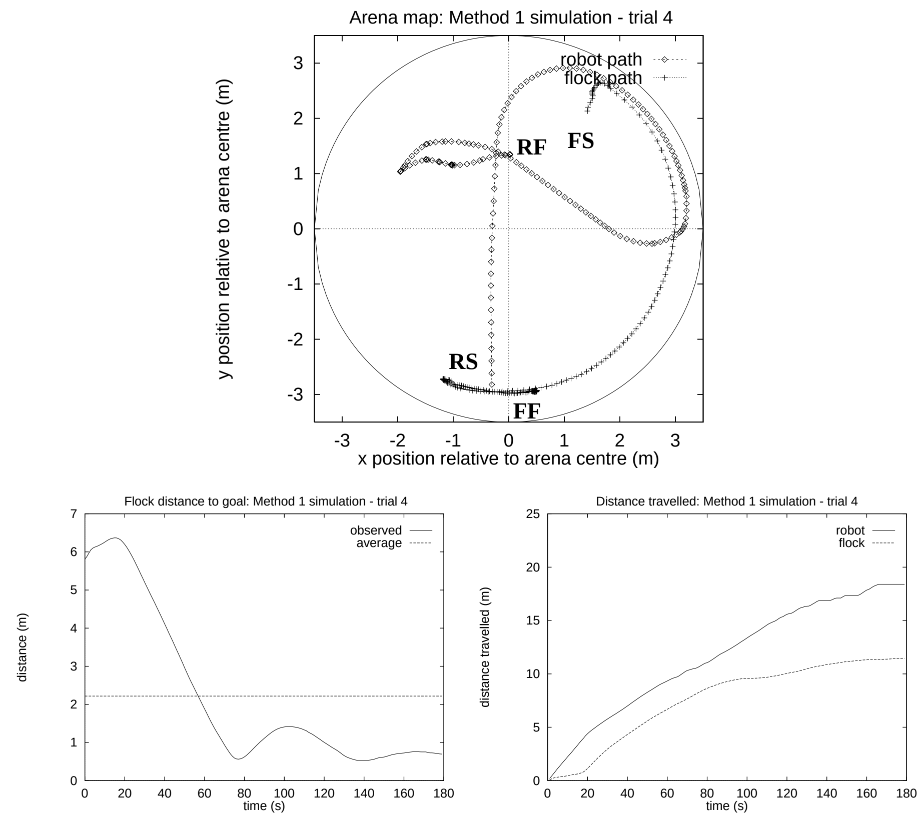


Figure 4.10: Method 1 simulation results - trial 4.

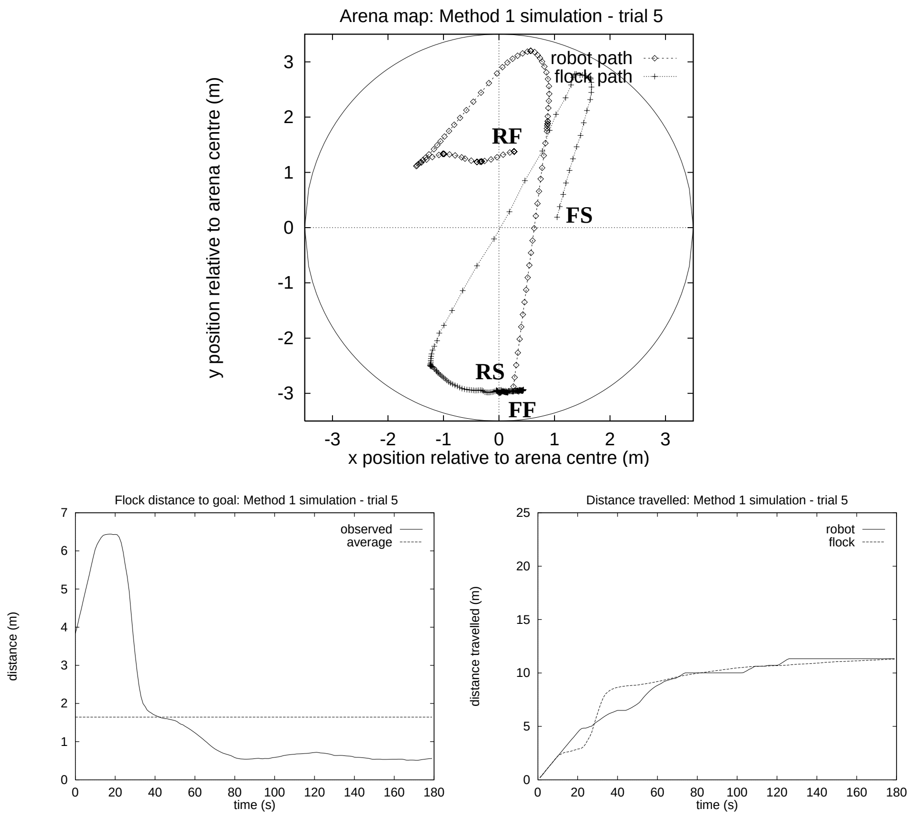


Figure 4.11: Method 1 simulation results - trial 5.

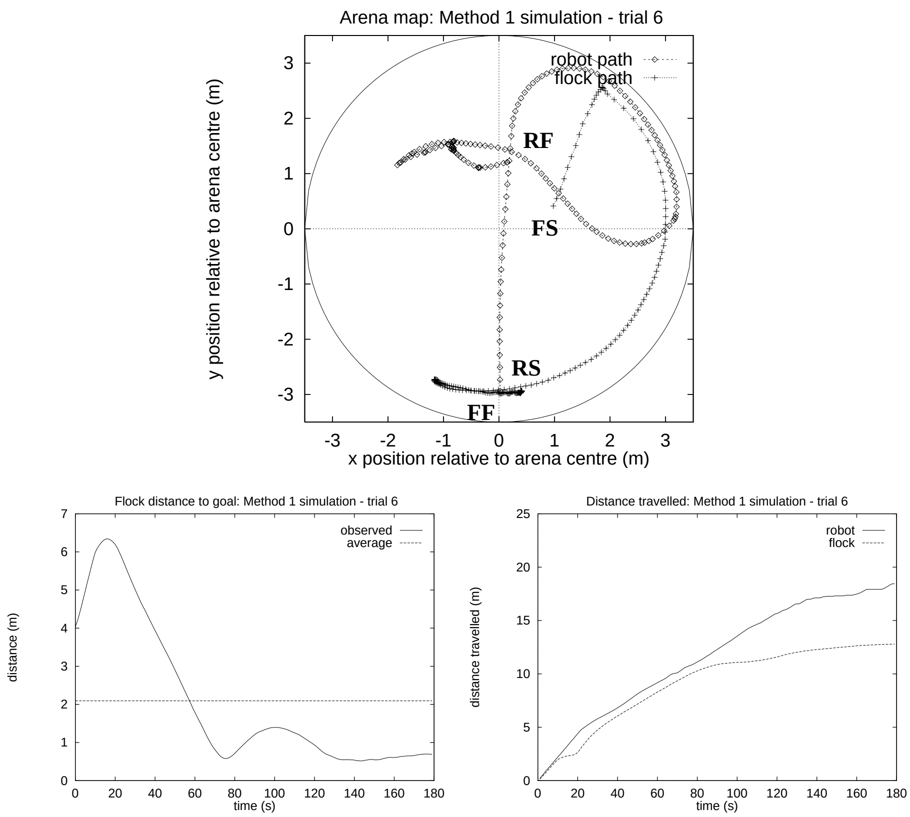


Figure 4.12: Method 1 simulation results - trial 6.

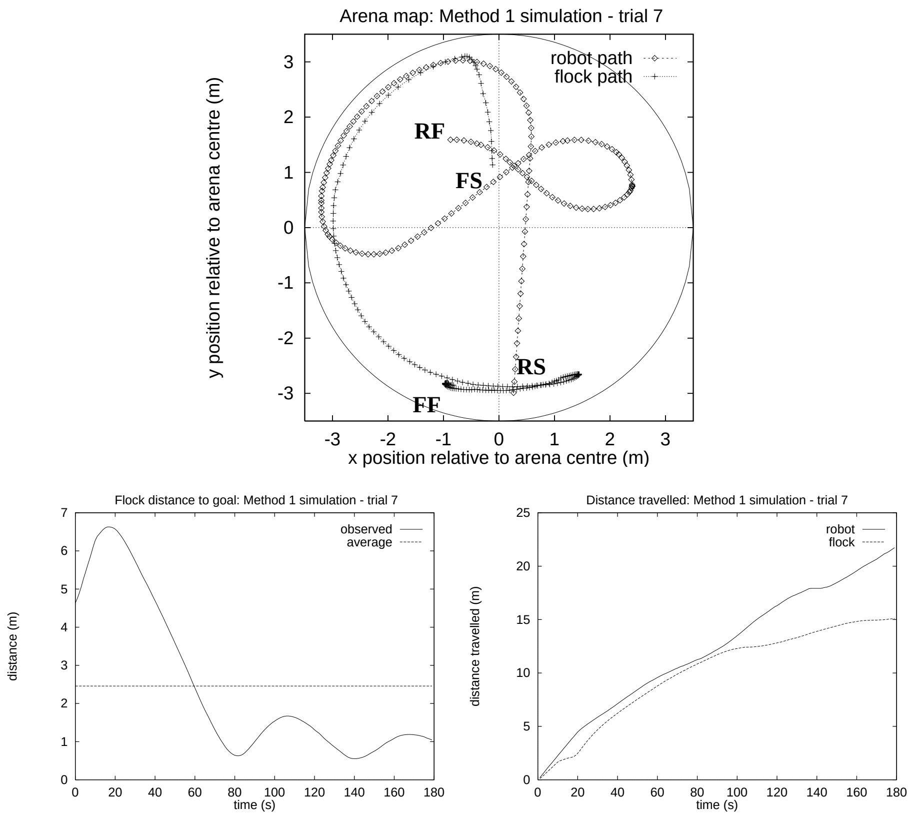


Figure 4.13: Method 1 simulation results - trial 7.

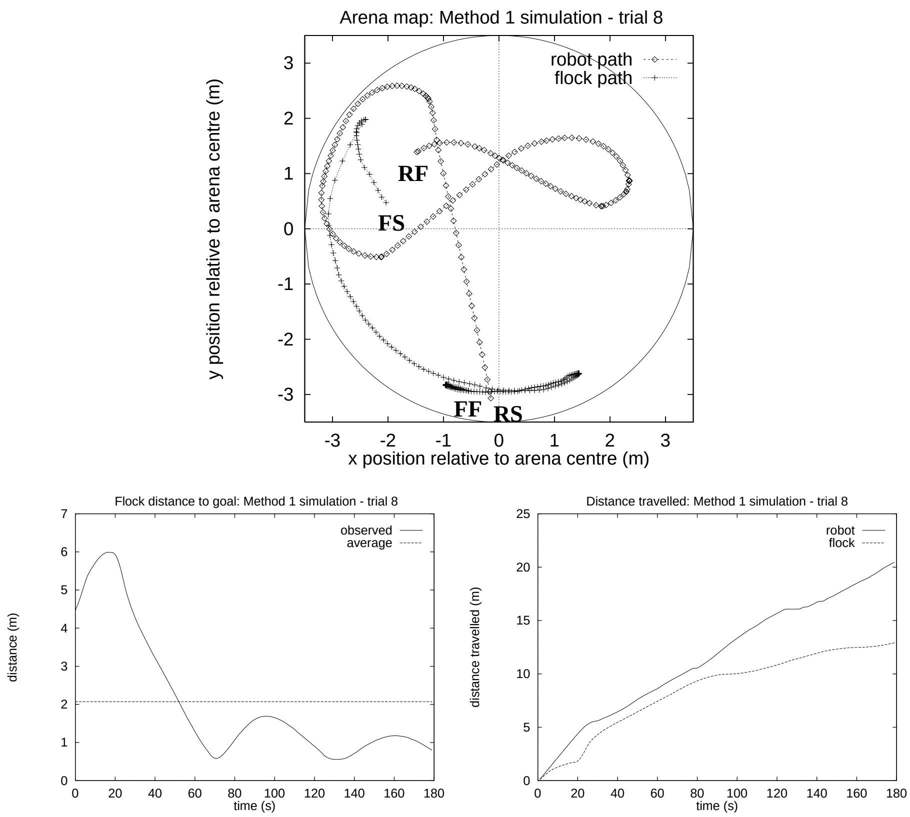


Figure 4.14: Method 1 simulation results - trial 8.

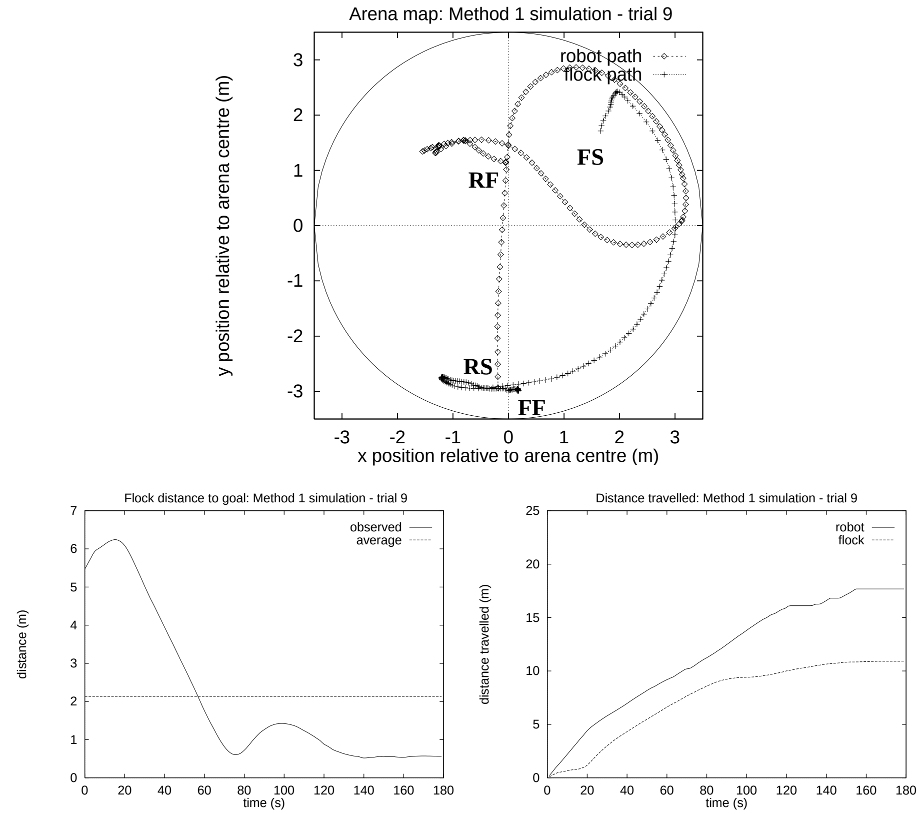


Figure 4.15: Method 1 simulation results - trial 9.

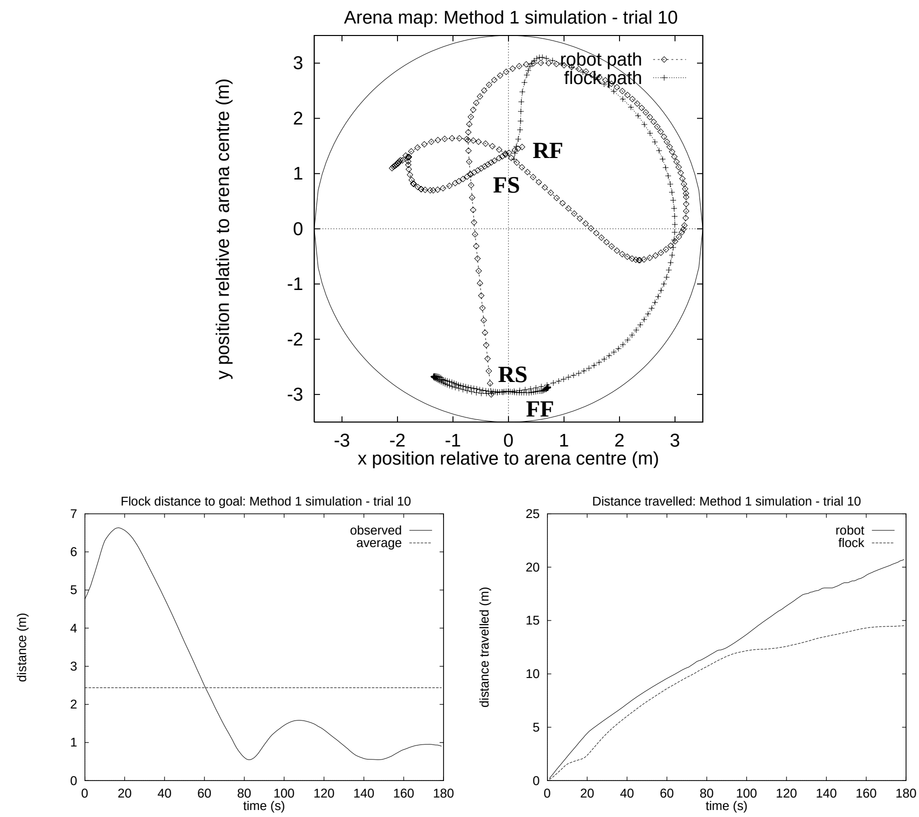


Figure 4.16: Method 1 simulation results - trial 10.

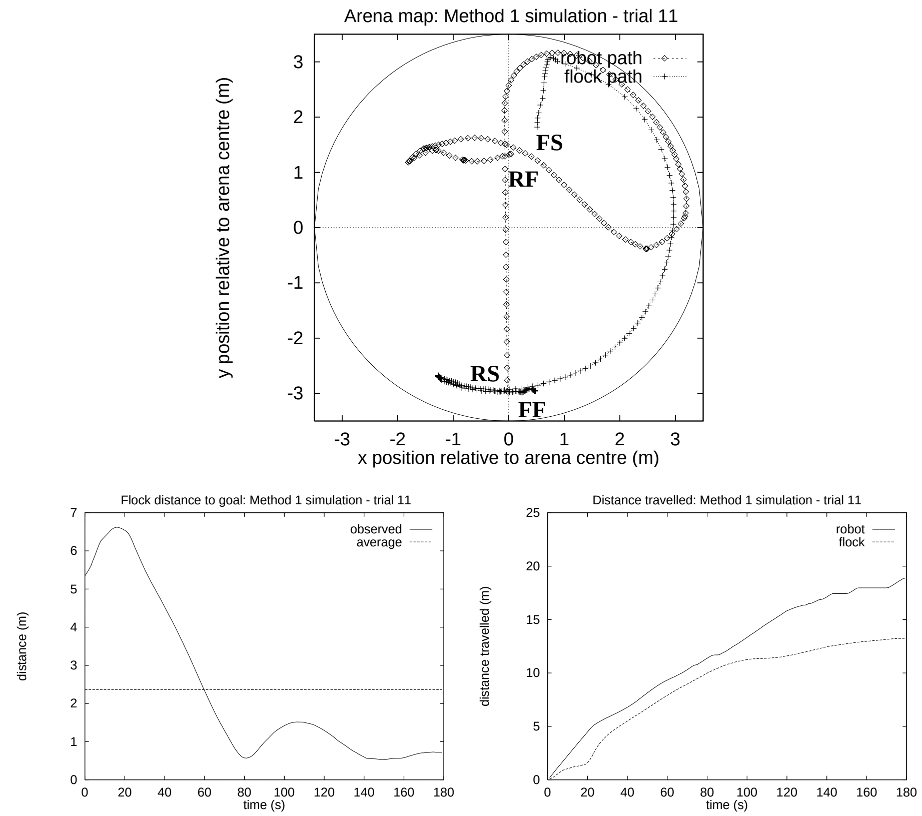


Figure 4.17: Method 1 simulation results - trial 11.

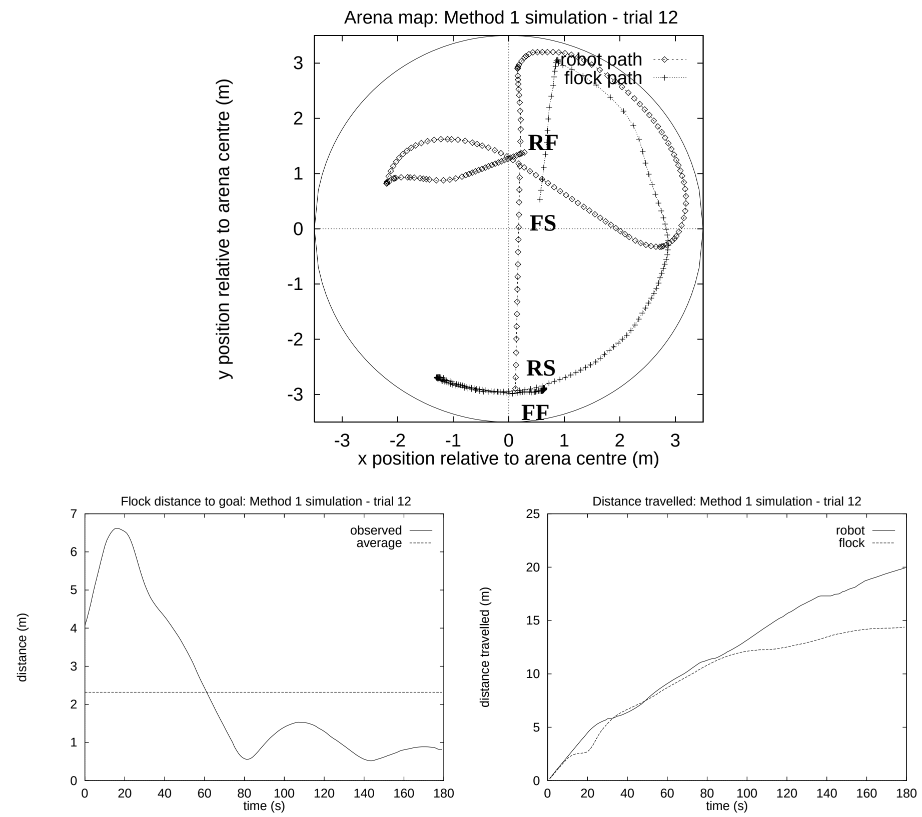


Figure 4.18: Method 1 simulation results - trial 12.

Trial	Efficiency			Success
	robot	flock	total	
1	21.49	12.57	34.06	2.23
2	17.11	13.42	30.53	2.22
3	20.82	12.83	33.65	2.11
4	18.40	11.47	29.87	2.22
5	11.33	11.30	22.63	1.64
6	18.44	12.79	31.23	2.09
7	21.73	15.10	36.83	2.46
8	20.42	12.91	33.33	2.07
9	17.68	10.92	28.60	2.13
10	20.72	14.52	35.24	2.44
11	18.84	13.24	32.08	2.36
12	19.90	14.38	34.28	2.32
total	226.88	155.45	382.33	27.00
average	18.91	12.95	31.86	2.19
variance	7.95	1.70	14.02	0.05
stdev	2.82	1.30	3.74	0.22

Table 4.1: Summary of results for Method 1 simulation trials. All results are measured in metres.

4.6 Real-world trials

4.6.1 Procedure

The simulation experiment was then repeated in the real world using the robot system and a real flock of ducks. A random point along the arena boundary is chosen as the flock goal. This point is changed for each trial so that the ducks cannot learn the task. With the robot inactive and positioned near the goal, a flock of 12 ducks is introduced into the arena. After 3 minutes accommodation time, the robot is activated. The positions of the robot and flock centre 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. A white breed was used to make the vision task easier.

The number of real-world trials was limited by consideration of the welfare of the experimental animals and by the considerable time required for each trial. Access to the ducks was also limited as they were subject to behaviour experiments by Jane Henderson.

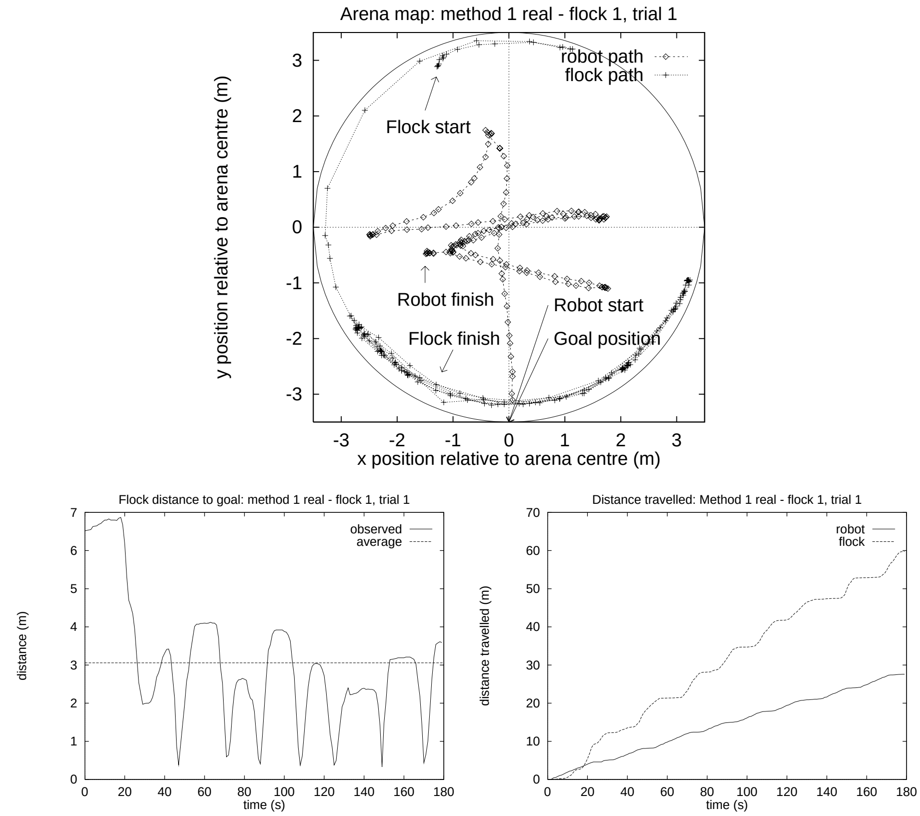


Figure 4.19: Method 1 real-world results - trial 1.

4.6.2 Results

Results are presented for each trial in Figures 4.19 to 4.27. A plot of the paths of the robot and flock centre through the arena is given (top). The path plots are labelled with the robot start position **RS**, robot finish position **RF**, flock start **FS**, and flock finish **FF**. The plots have been rotated so that the goal position is always at the bottom of the arena at (0,-3.5).

Also given are plots of the flock distance-to-goal (left) and the distance traveled by the robot and the flock (right). These data are used to calculate final success and efficiency scores for each trial. These scores are presented in Table 4.2 along with their averages and variances.

The average success score over 9 trials for Method 1 in the real world was 2.65m. The average efficiency score over 9 trials was 70.33m, of which 43.24m was flock movement. The variance of both metrics was larger in the real world than in the simulation.

Trial 1 is representative of the moderately successful real-world trials. Figure 4.19 (top) shows that its behaviour is similar to that seen in the simulation trials. The flock is initially pushed

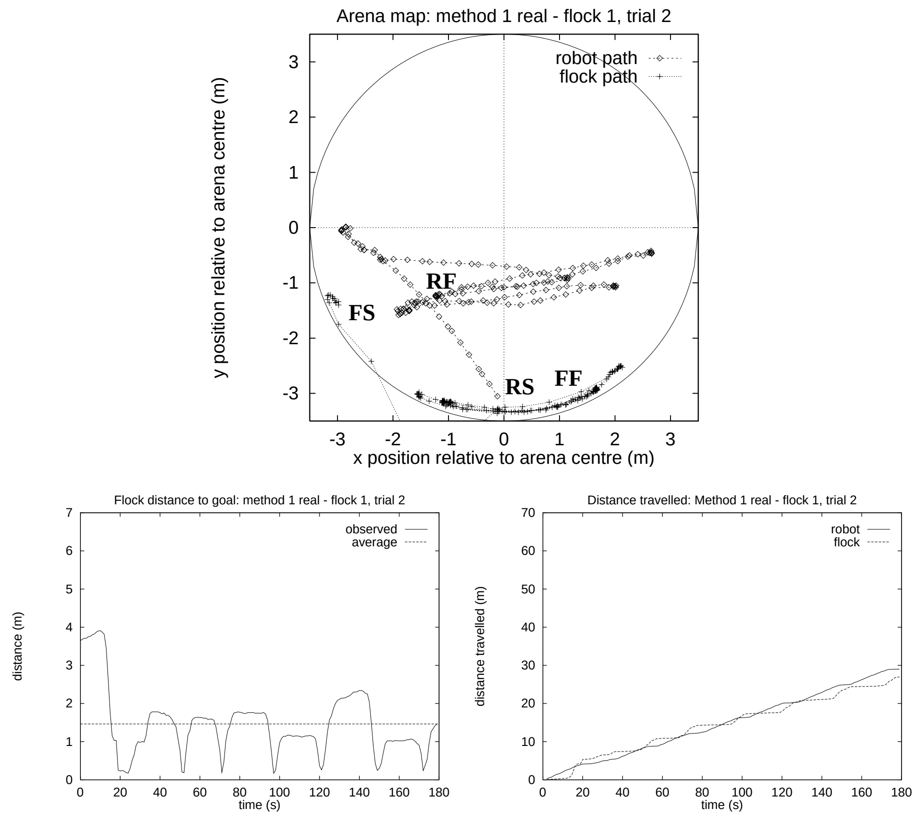


Figure 4.20: Method 1 real-world results - trial 2.

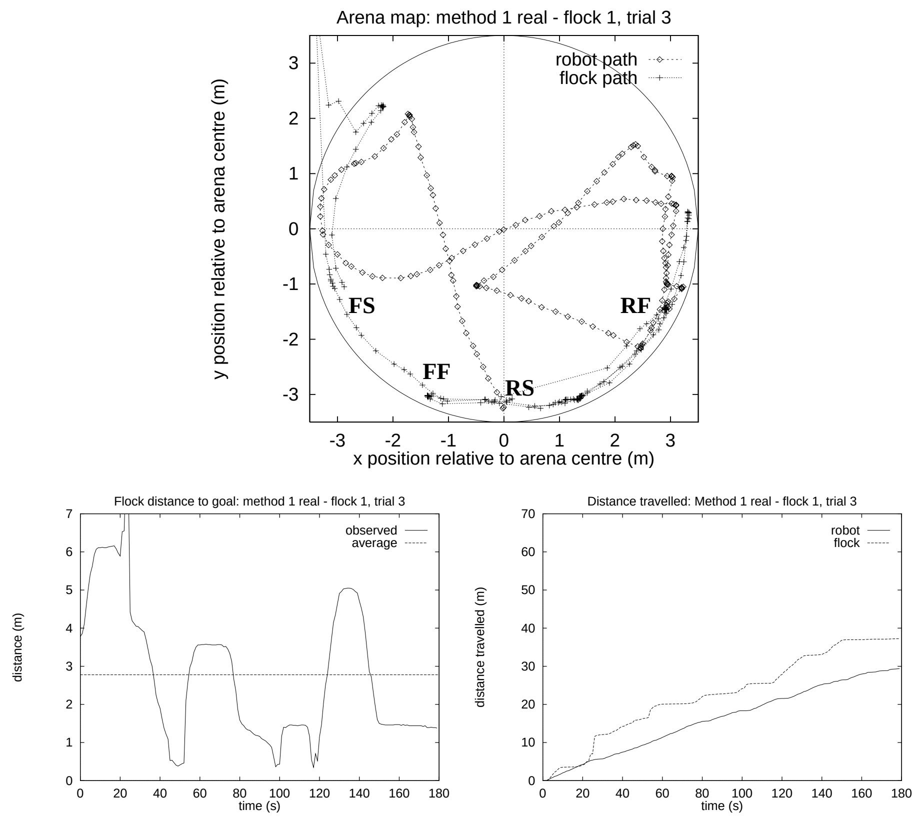


Figure 4.21: Method 1 real-world results - trial 3.

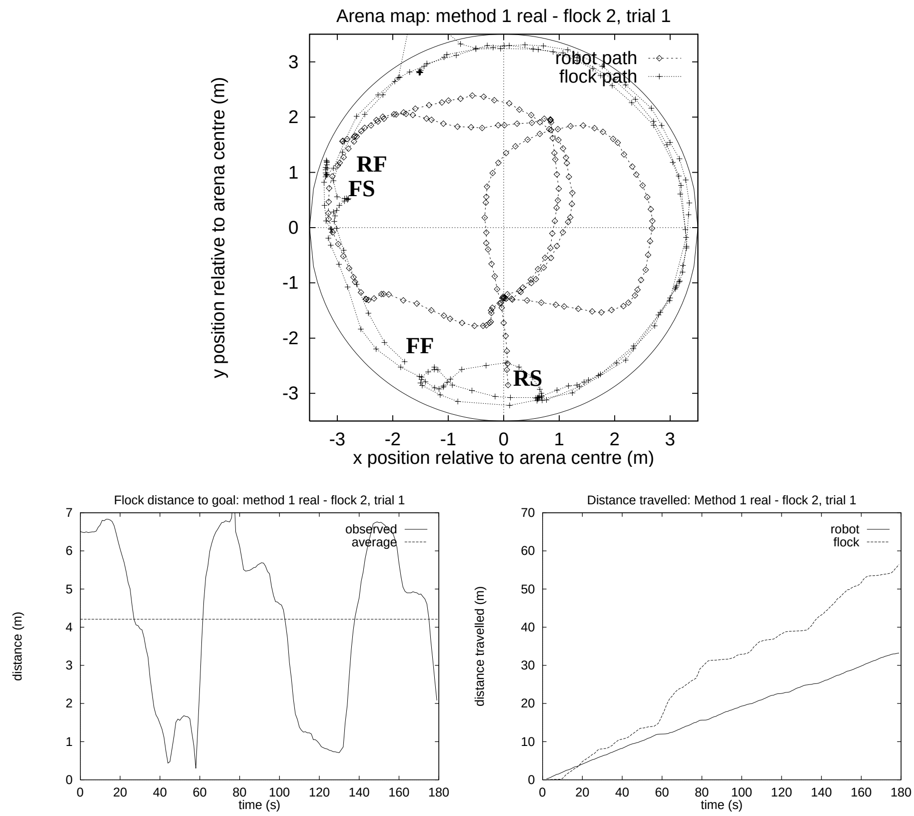


Figure 4.22: Method 1 real-world results - trial 4.

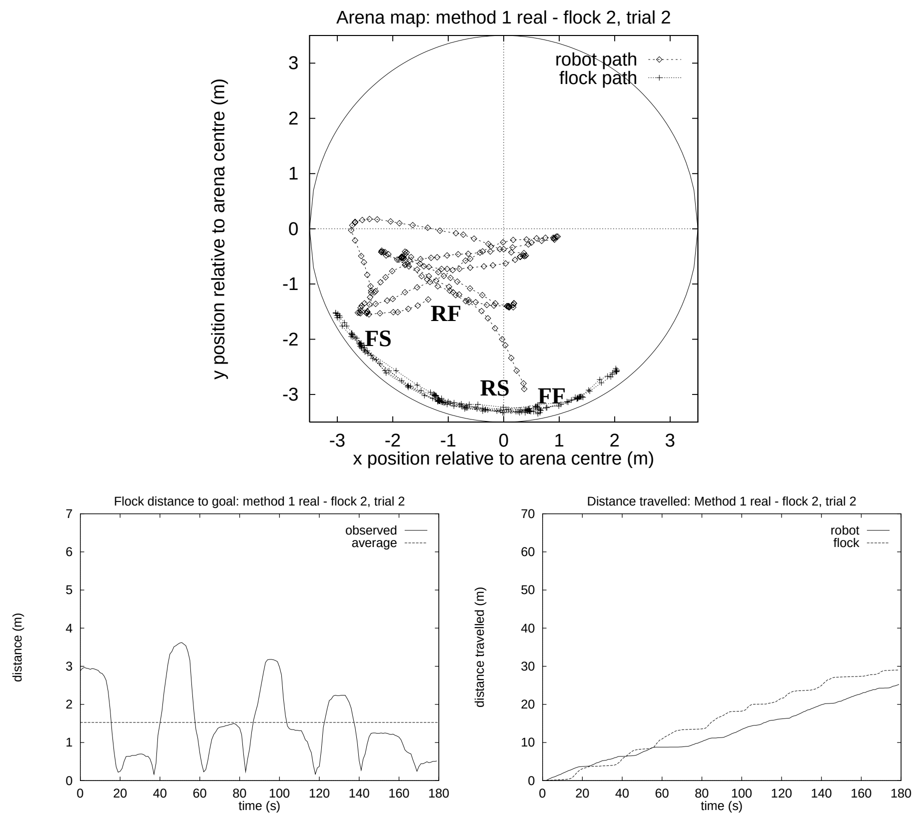


Figure 4.23: Method 1 real-world results - trial 5.

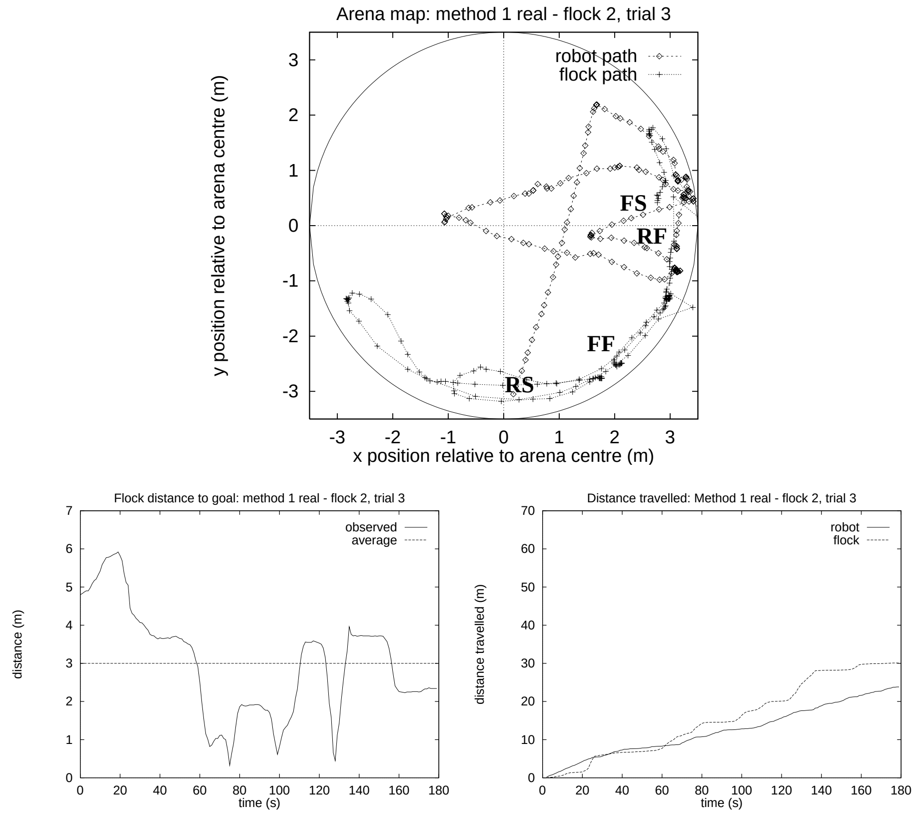


Figure 4.24: Method 1 real-world results - trial 6.

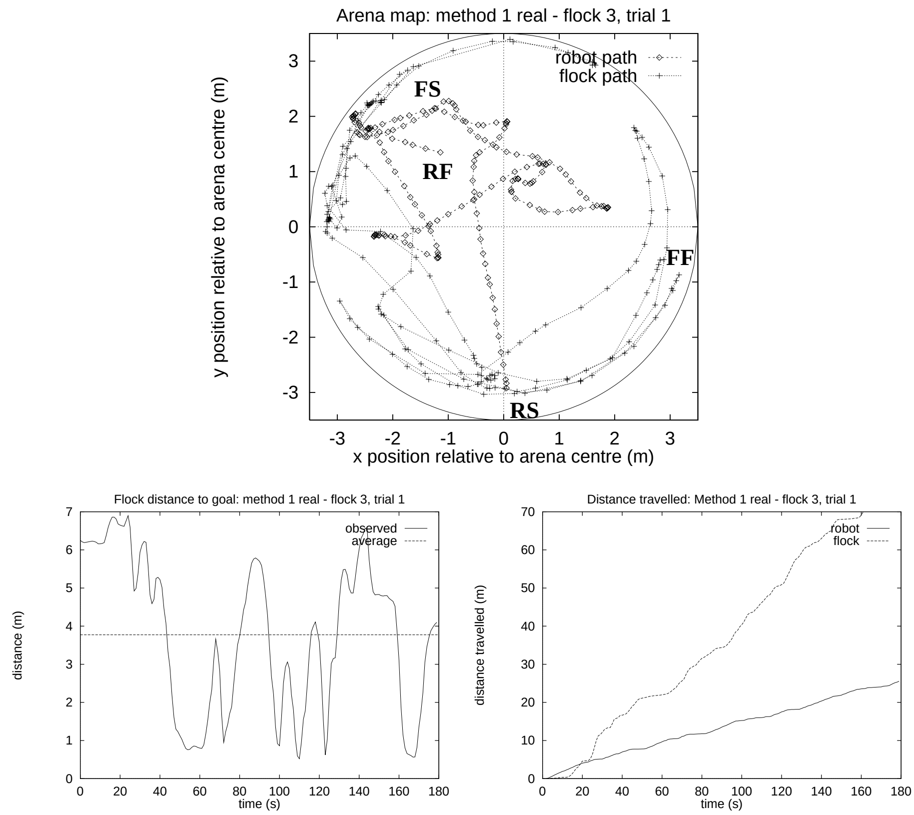


Figure 4.25: Method 1 real-world results - trial 7.

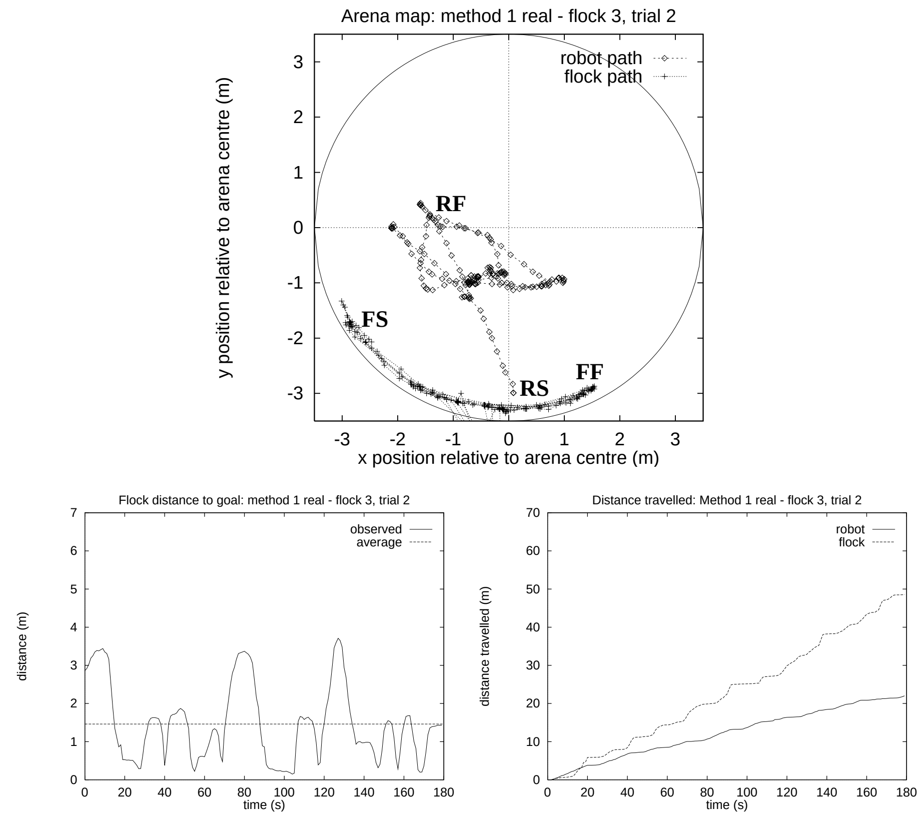


Figure 4.26: Method 1 real-world results - trial 8.

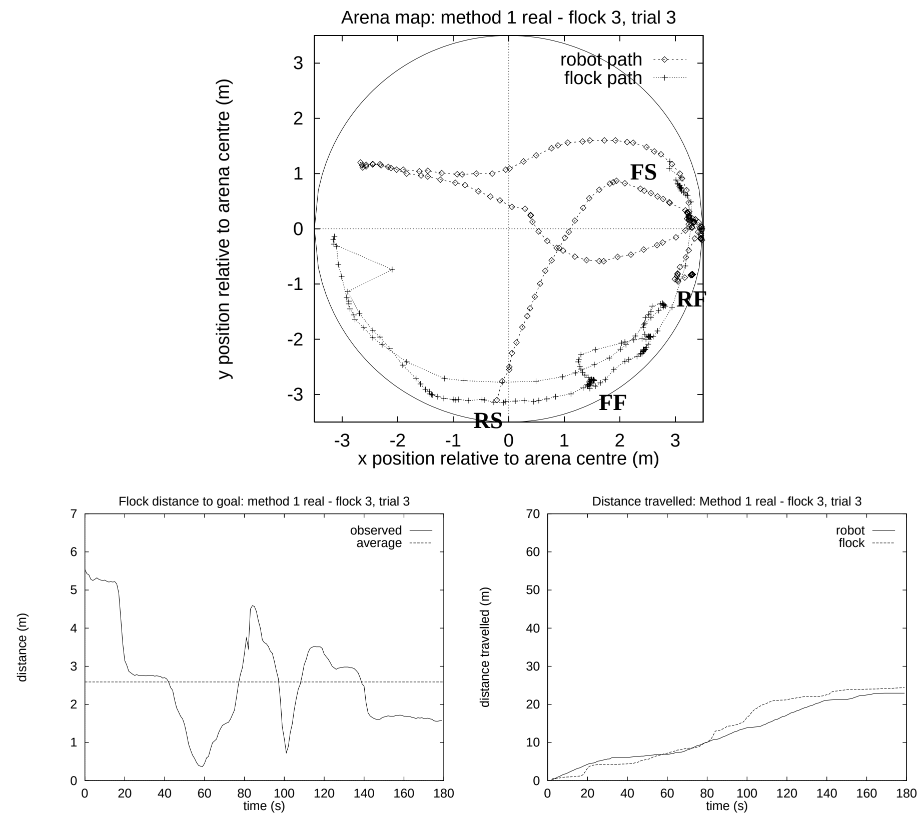


Figure 4.27: Method 1 real-world results - trial 9.

Trial	Efficiency			Success
	robot	flock	total	
1	27.59	59.76	87.35	3.06
2	29.01	26.98	55.99	1.46
3	29.38	37.23	66.61	2.78
4	33.24	56.46	89.70	4.21
5	25.24	29.01	54.25	1.53
6	23.82	30.01	53.83	3.00
7	25.56	76.80	102.36	3.77
8	22.02	48.52	70.54	1.46
9	27.95	24.37	52.32	2.59
total	243.81	389.14	632.95	23.86
average	27.09	43.24	70.33	2.65
variance	11.24	329.49	345.69	1.01
stdev	3.35	18.15	18.59	1.01

Table 4.2: Summary of results for Method 1 real-world trials. All results are measured in metres.

away from the goal until it reaches the arena wall, then pushed down the wall towards the goal by the robot. Then the flock oscillates around the goal, fetched back each time by the robot. The movement of the ducks, and hence the frequency of oscillation is much faster than in the simulation, due to the timescale differences discussed above. More significantly, the amplitude of the oscillations is higher than in the simulations, producing a worse success score. The success plot Figure 4.19 (left) clearly shows the oscillatory behaviour.

4.6.3 Discussion

All parameters in the Method 1 controller were identical to those used in the simulations, so the difference in behaviour must be due to to the dynamics of the real-world system. It is likely that the most significant of these is the time delay in the control loop. The simulation assumes zero delay in the vision system, but in the real robot there is at least a 60ms delay in processing the camera images. More significantly, the simulation has idealized robot dynamics; the vehicle's wheel speeds match those requested by Method 1, so the vehicle very closely approximates the path generated by the algorithm. The proportional controller in the real vehicle inevitably has a limited frequency response to incoming wheel speed requests. The real vehicle is also subject to inertia, friction, etc., none of which are modeled.

Improving the accuracy of the robot and/or flock models might allow better off-line parameter choices. Indeed, it was found by experiment with the simulation that the size and duration of these oscillations can be reduced by careful parameter tuning. However, as argued above, the variation in behaviour between different species, different flocks, or even the same flock over time may be subject to large variation. As the goal of this work is to produce a *general* flock control strategy, and to investigate general robot/animal interaction, then it is required to keep the models as simple and general as possible. Robustness, including minimal parameter sensitivity, should be an intrinsic property of the algorithms developed.

It will be useful later to define a condition whereby the system can be said to have 'worked'. Considering the real-world results, it was determined that trials 1, 2, 5 & 8 worked in the subjective sense that the behaviour was similar to the simulation, and in the objective sense that the ducks were brought near the goal. However, in trial 1 the distance that the flock oscillated around the flock was unacceptably large, producing a success score of 3.06m. This is only 23% better than the the average score achieved by a random distribution of ducks in the arena ($\approx$ 3.96m), so this trial cannot be said to have worked. The success scores of trials 2, 5 & 8 were all below 2m and all the non-working trials have success scores above 2m. A success score of <2m is therefore adopted as the test for a real world working system.

Note that the Method 1 simulated trials generally score just over 2m success, but as they are clearly otherwise successful, this can be considered an effect of an uneven timescale. The simulated trials take around 80s to bring the flock close to the goal for the first time, but the real trials take about 40s. Thus a much larger proportion of the 180s trial is taken up by the initial fetching phase, reducing the overall success score. As the robot moves in response to the movements of the ducks, this difference can be explained by the faster movement of the real ducks in response to the vehicle.

To obtain a fairer comparison of simulated and real success scores the ducklets' parameters could be adjusted to more closely match the behaviour of the real-world ducks. Alternatively, since the time in the simulation is arbitrary, the timescale could be scaled such that (for example) the first goal-crossing happened at the same time as in the real world. This is not considered important, however, because the model ducks will never exactly match the real ducks, and in any case the ability of the system to be successful in dealing with widely differing flocks is very desirable. This difference in timescale should be borne in mind when comparing the simulated and real results. The <2m criterion applies only to the real world system.

Method 1 worked in 33% (3/9) of these trials.

4.7 Conclusions

(1) It was shown that a robot system can exploit the threat-avoidance behaviour of a flock in order to control its movement (Hypotheses 1 & 2).

(2) A flock-control algorithm was developed using only a generalized simulation of flocking behaviour, yet it transferred unchanged to the real flock (Hypothesis 3). However, performance was worse and varied more in the real world than with the simulated flock.

(3) This experiment was the first demonstration of a robot system that can control the behaviour of an animal to achieve a useful task. However, robustness in the real world was poor ($\ll$50% success rate).

4.8 Further work

Further informal 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 4.1). This effectively controls the distance that the robot keeps from the flock, which is related to the 'push' exerted on the ducks to control their movement. The success of the system is sensitive to changes in this parameter. The large variation likely to be found 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 achieved with an appropriate adaptive algorithm, but consideration of these results suggested another, simpler flock control algorithm which solves this problem and is described in the following chapter.

Chapter 5

Flock control 2

The previous chapter showed the design of the Method 1 flock control algorithm in simulation, and its its direct transfer to the real world. Though the method succeeded in gathering the simulated ducks close to the goal position, Method 1 did not work reliably in the real world, and it produced an undesirable oscillatory movement of the flock around the goal point. This chapter presents an improved flock control algorithm, Method 2, which is assessed in a set of similar experiments.

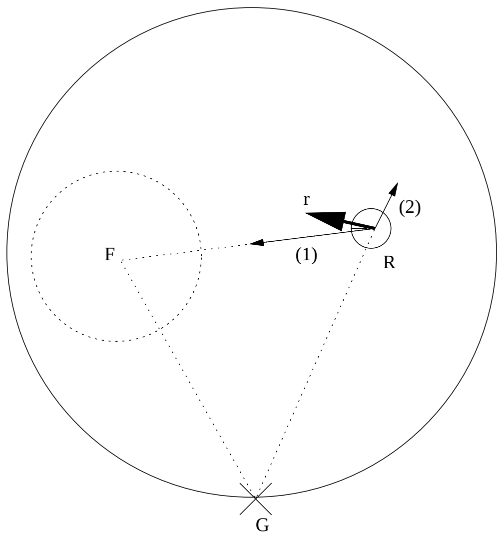
Work presented in this chapter has been published as refereed conference papers [Vaughan et al., 1998a] [Vaughan et al., 1998b] [Vaughan et al., 1998c].

5.1 Hypothesis

The distance from the flock to the goal $|GF|$ (see Figures 4.1 & 5.1) is the system variable to be controlled, ie. reduce to zero. In a classical proportional controller a control signal would be applied to correct this variable, with a magnitude proportional to the size of the error. If this term is introduced into the flock controller, an analogous system can be designed 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.

5.2 Algorithm

In the Method 2 flock control algorithm, the robot's movement vector $\vec{r}$ is given by the function shown in Figure 5.1. 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. This algorithm is simpler than the previous version, with just two terms and two parameters.



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

Figure 5.1: Method 2 (schematic not drawn to scale). Key: gain parameters $K_{1,2}$; flock centre F; Robot position R; Goal position G; algorithm terms ($1 \rightarrow 3$) and resultant $\vec{r}$ (where $\widehat{a}$ is the unit vector of $\vec{a}$)

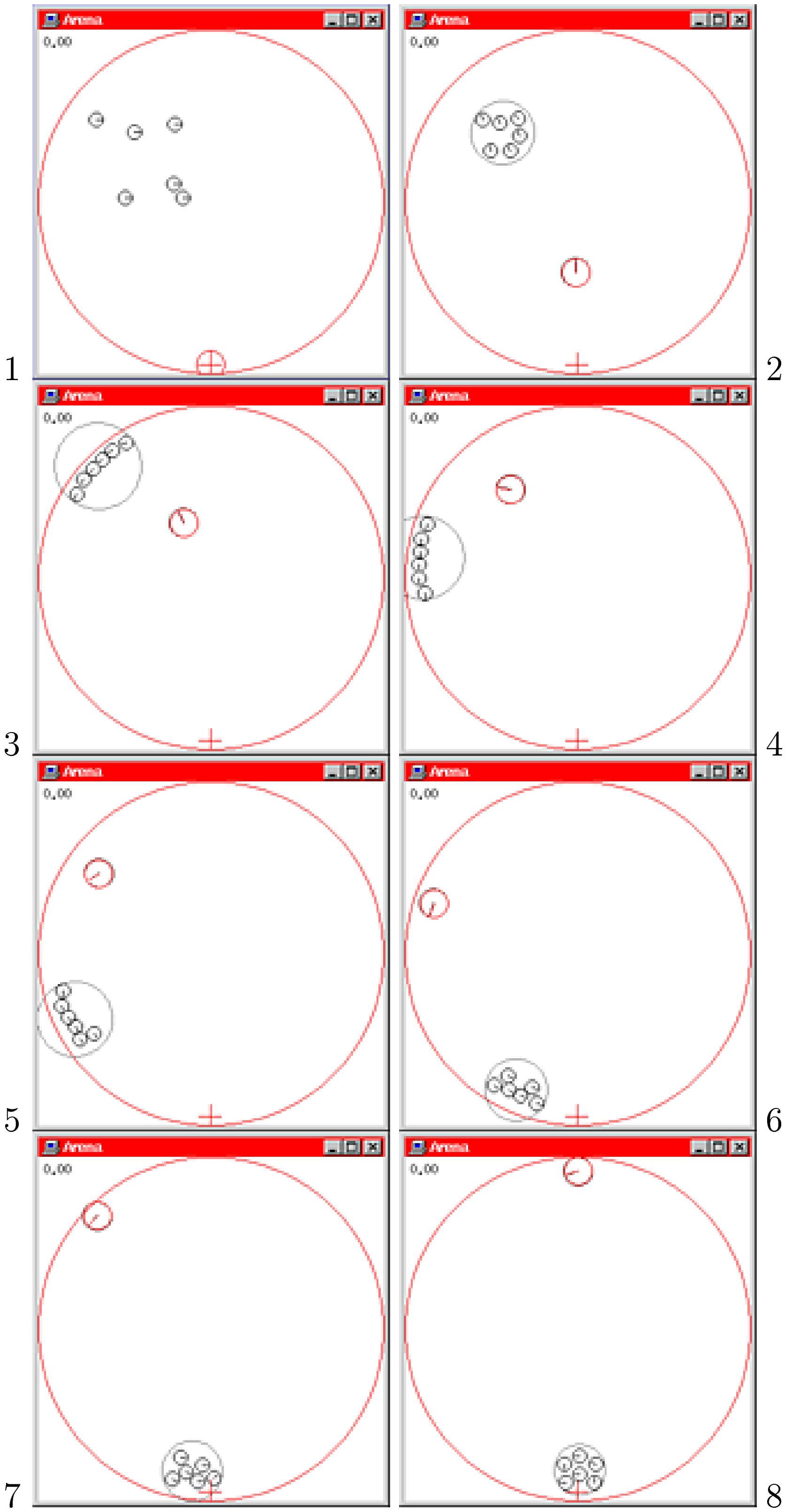
5.3 Simulation trials

5.3.1 Procedure

The experimental procedure for the simulation trials was identical to that described for Method 1 trials as described in the previous chapter. The same data were recorded from each trial, and are presented below.

5.3.2 Example simulation trial

Figure 5.2 shows screenshots of Trial 1 as it ran. The robot starts at the goal position at the bottom of the arena, with the ducklets loosely clustered in the top left (1). As the trial starts, the robot is attracted to the flock and repelled from the goal, so it moves towards the right hand side of the flock, which aggregates and moves away from the robot (2). As the ducklets reach the edge of the arena and the robot moves around behind them (3), they begin to move along the arena wall towards the goal (4). As they move down the arena, the robot is pulled towards them by the flock attraction potential, maintaining the pressure on the ducklets to move towards the goal (5). As the flock-goal distance decreases, the robot's flock-attraction begins to reduce, and the distance the robot keeps from the flock increases, thus reducing the pressure on the flock (6). With the



Key: = robot, = ducklets, + = flock goal

Figure 5.2: Sequence of images from the simulator during a trial, showing successful behaviour.

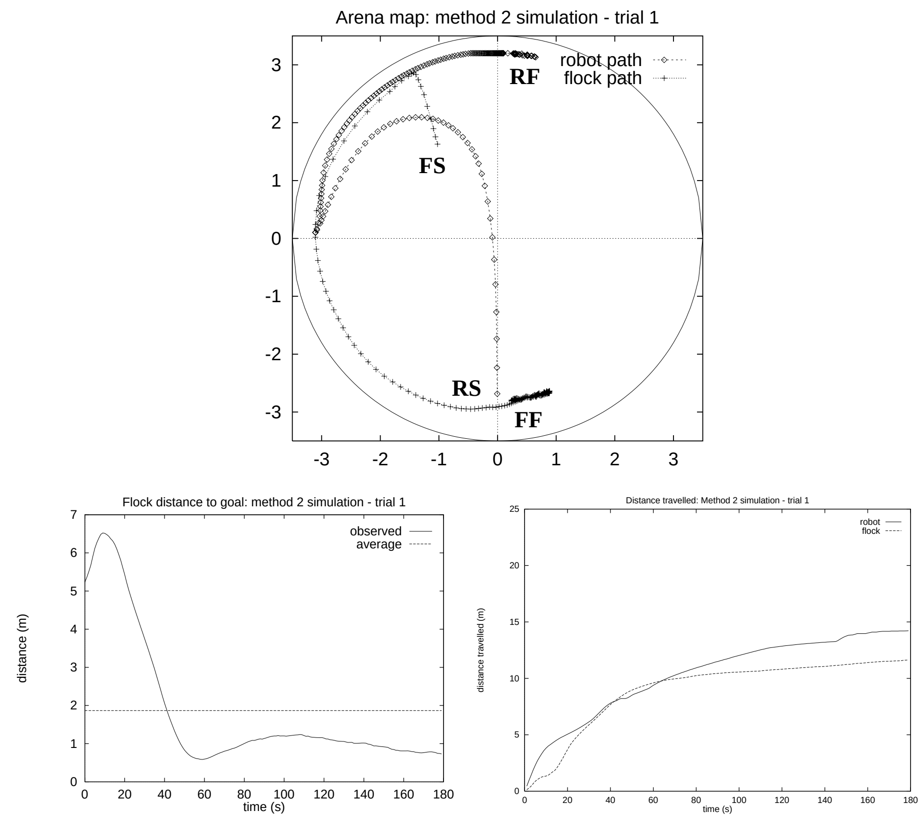


Figure 5.3: Method 2 simulation results - trial 1.

flock close to the goal, the robot's goal-repulsion dominates, and it is no longer attracted to the flock (7). Thus it seeks the furthest point from the goal, minimizing the stimulus to the ducklets (8). The system stabilizes with the ducklets at the goal and the robot on the far side of the arena.

The success plot of Figure 5.3 (left) shows that the oscillatory behaviour of Method 1 is greatly reduced (compare with Figure 4.7). Some overshoot of the goal occurs, but the stimulus to the ducks is quickly reduced by moving the robot away. Once the robot has reached the far side of the arena, there is no longer any sideways pressure on the ducks, so the oscillations are reduced to zero.

The path plot Figure 5.3 (top) shows the characteristic shape of the robot and flock paths that are evident with slight variation in all the other Method 2 simulation trials.

5.3.3 Results

Results are presented for each trial in Figures 5.3 to 5.14. A plot of the paths of the robot and flock centre through the arena is given (top). The path plots are labelled with the robot start

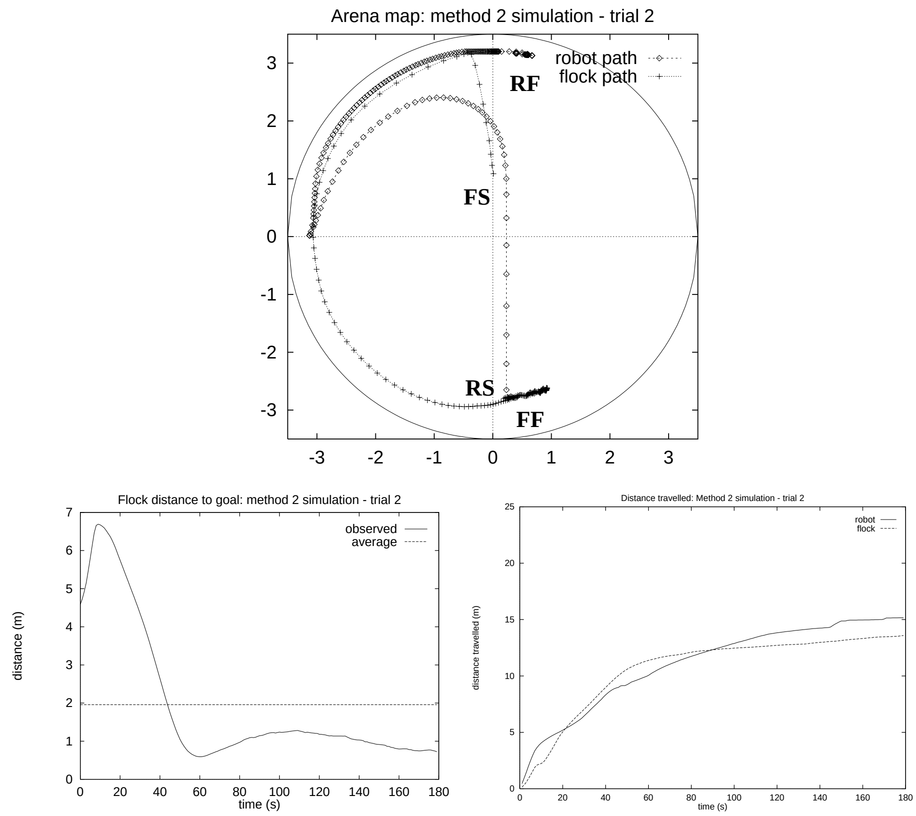


Figure 5.4: Method 2 simulation results - trial 2.

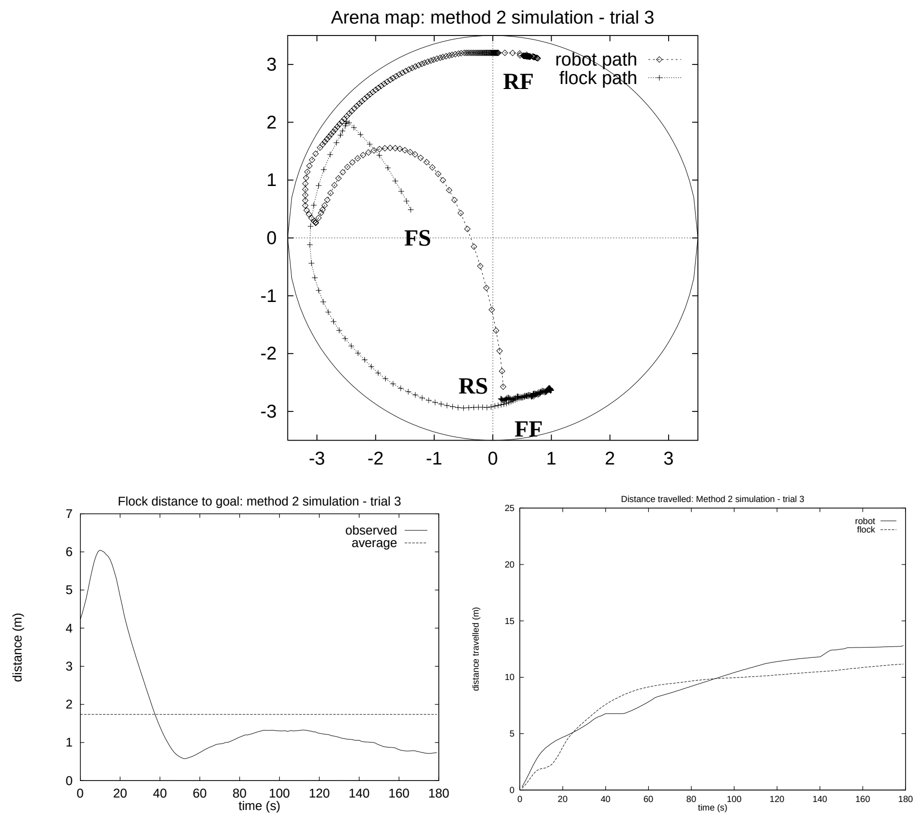


Figure 5.5: Method 2 simulation results - trial 3.

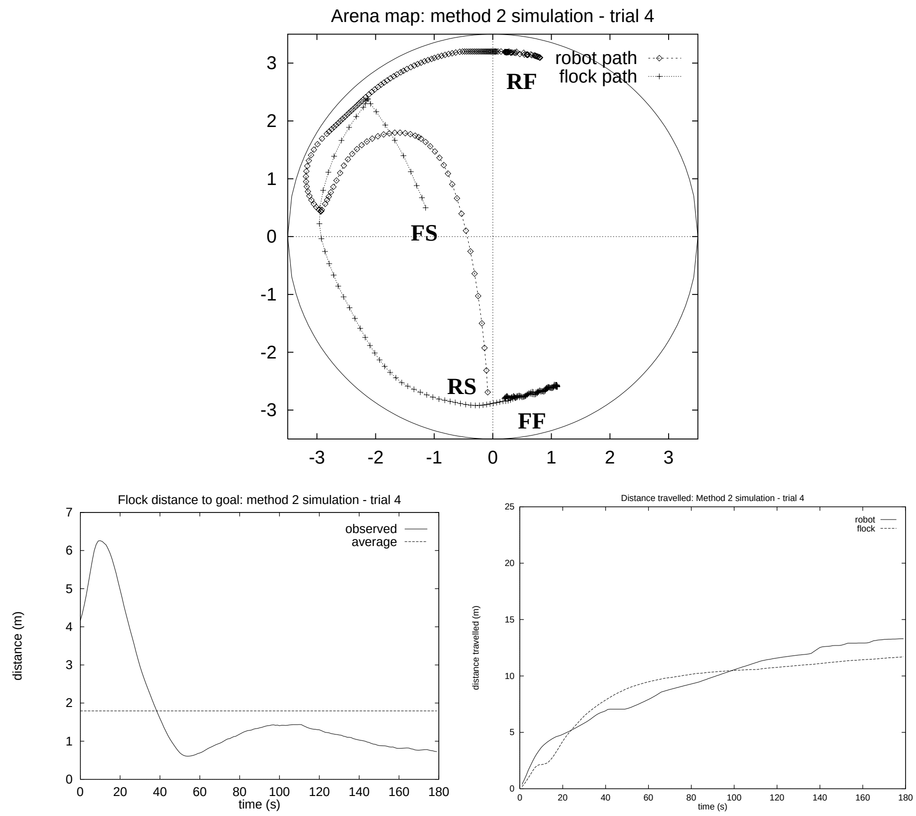


Figure 5.6: Method 2 simulation results - trial 4.

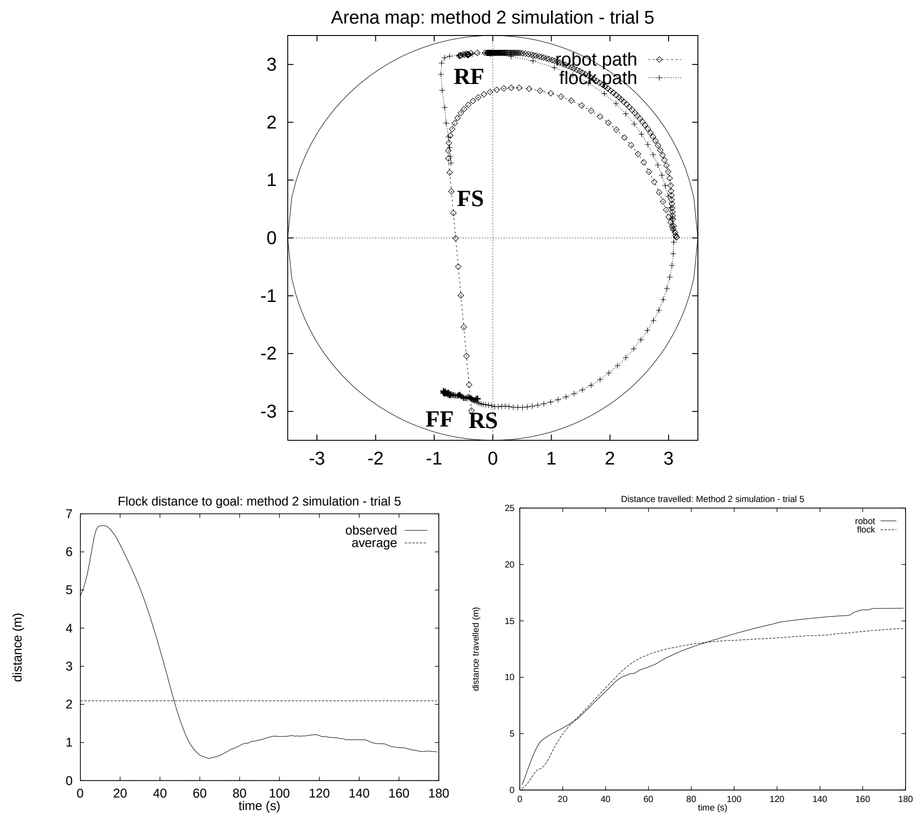


Figure 5.7: Method 2 simulation results - trial 5.

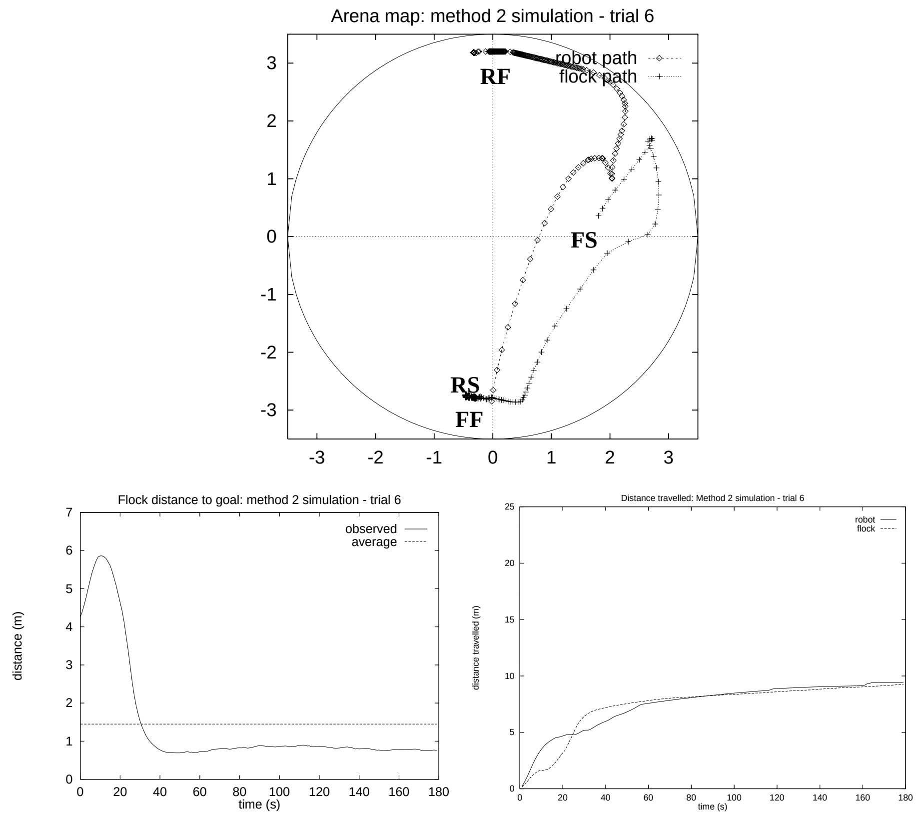


Figure 5.8: Method 2 simulation results - trial 6.

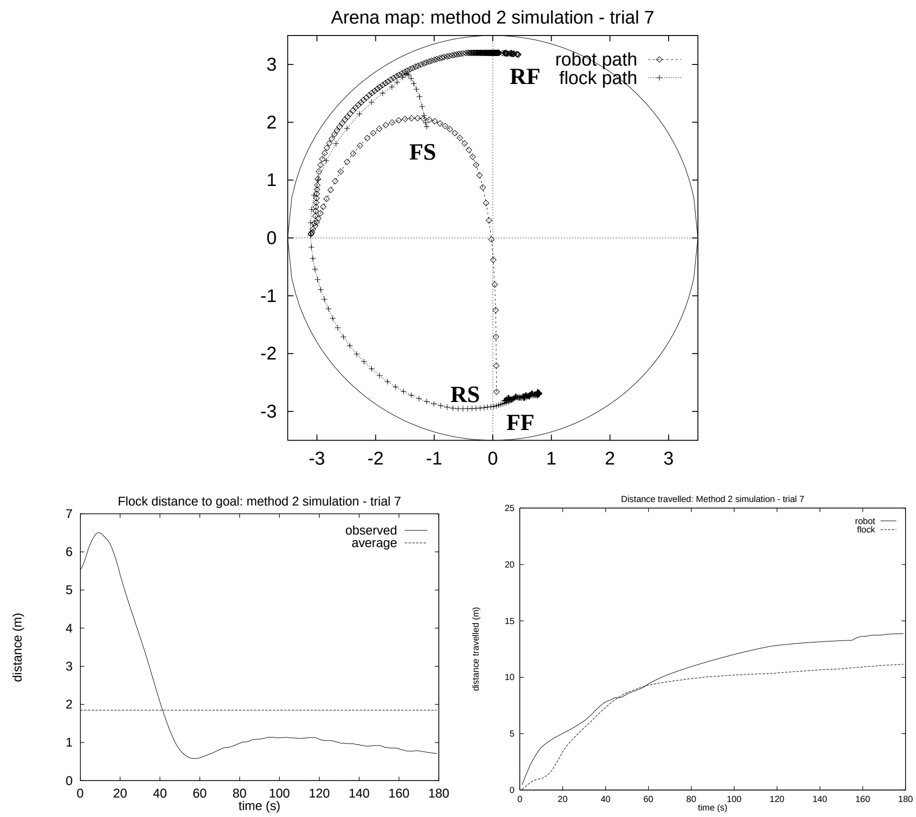


Figure 5.9: Method 2 simulation results - trial 7.

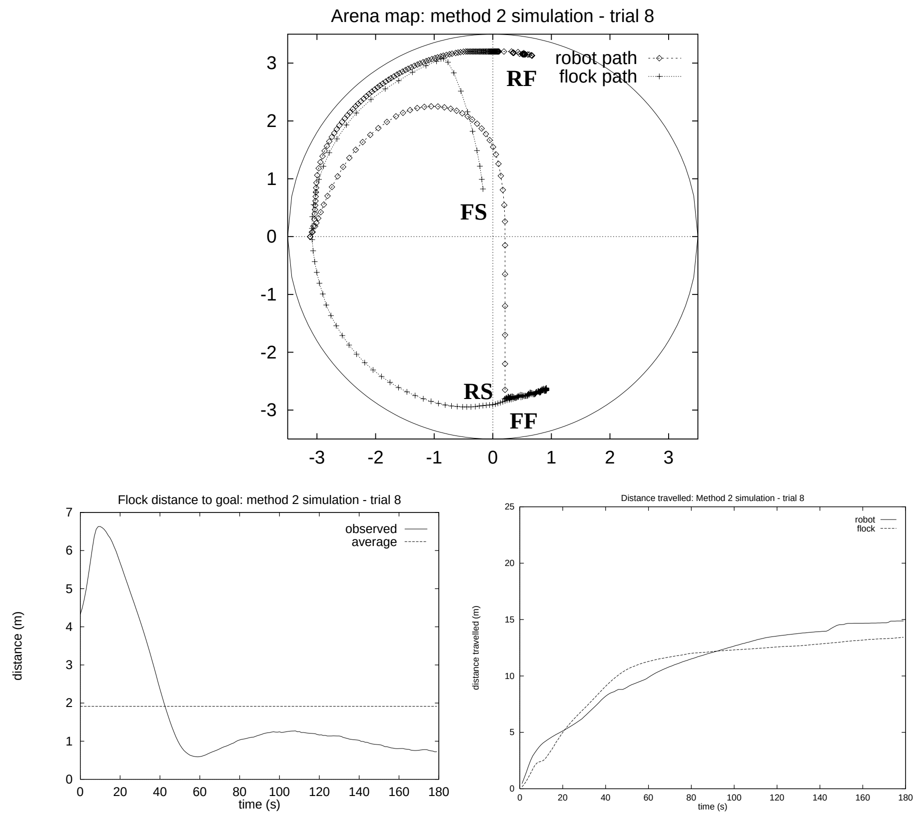


Figure 5.10: Method 2 simulation results - trial 8.

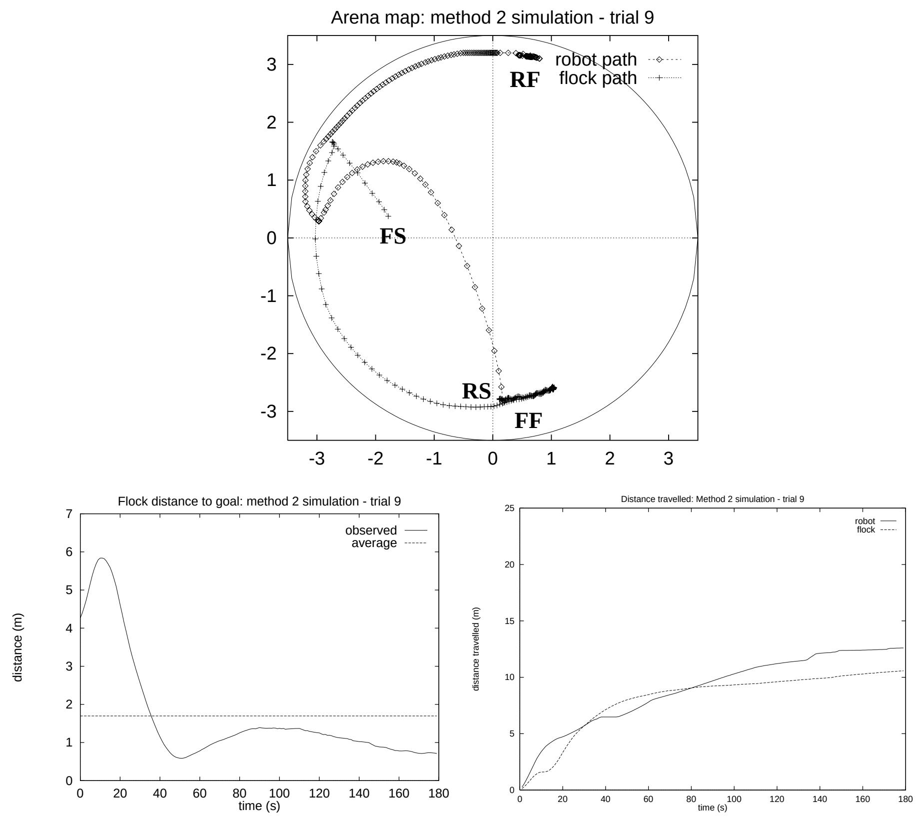


Figure 5.11: Method 2 simulation results - trial 9.

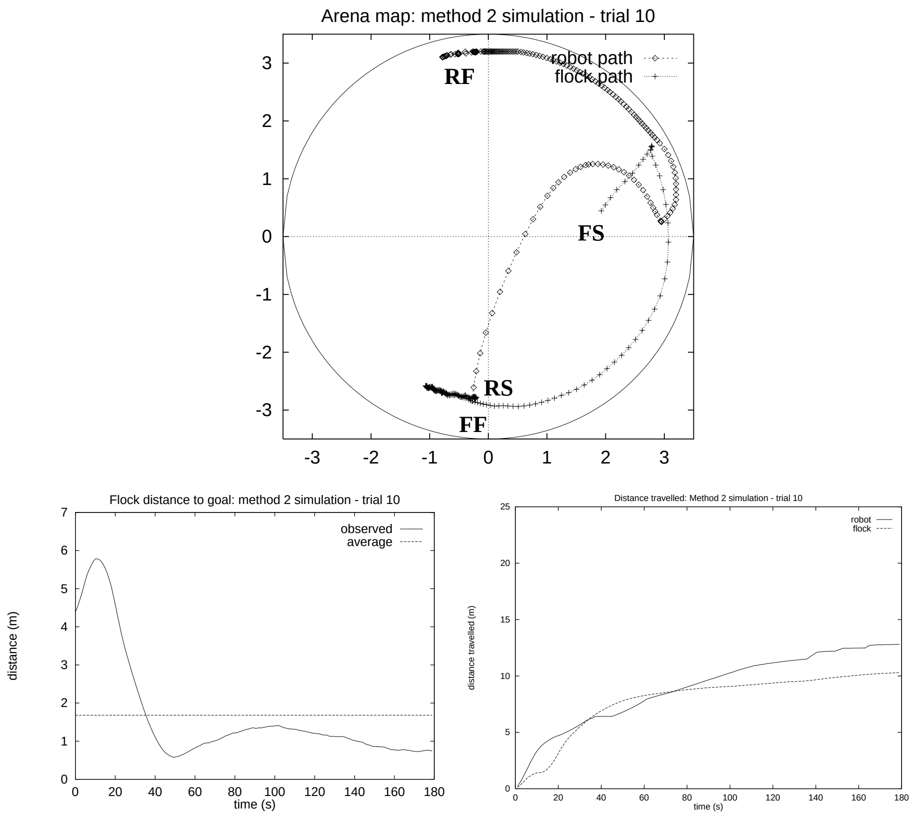


Figure 5.12: Method 2 simulation results - trial 10.

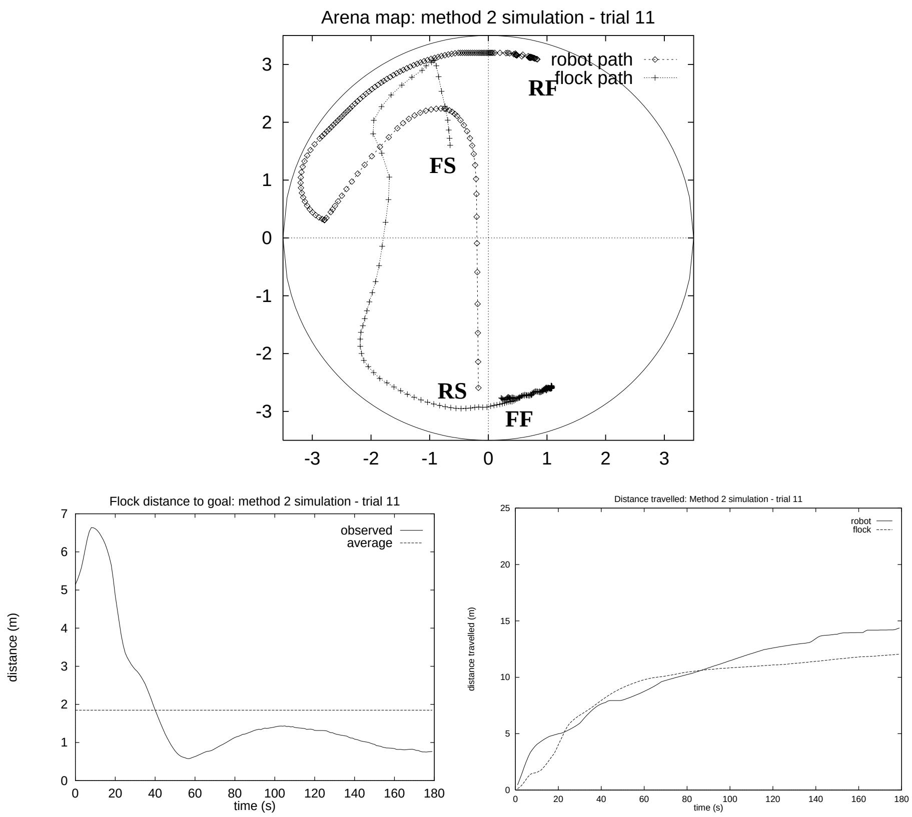


Figure 5.13: Method 2 simulation results - trial 11.

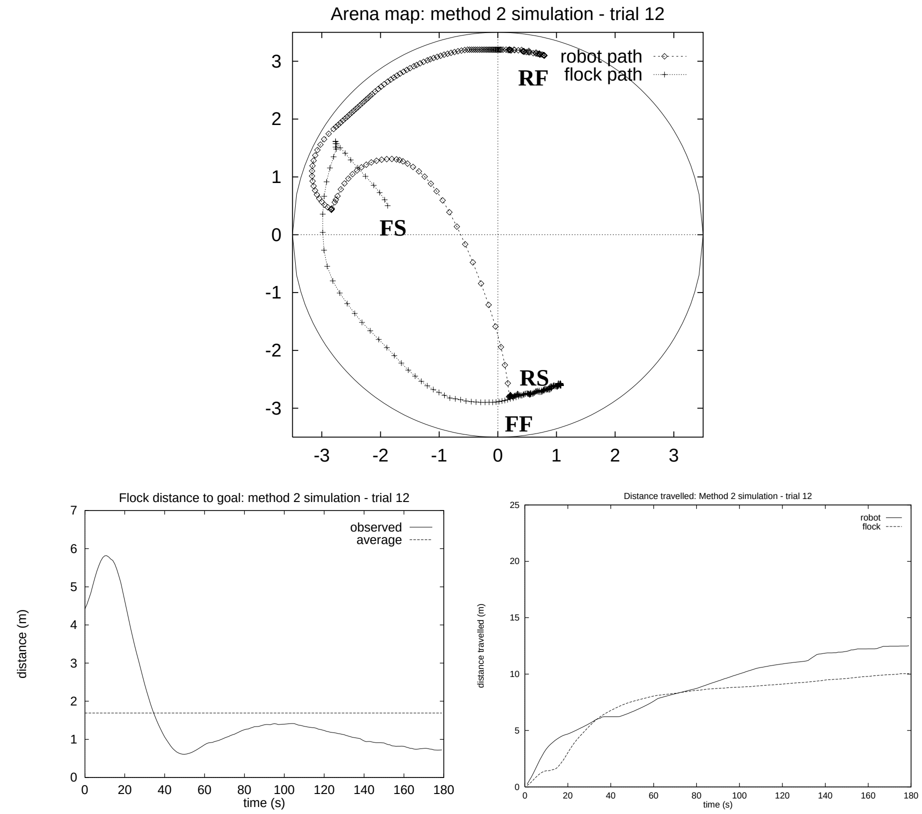


Figure 5.14: Method 2 simulation results - trial 12.

Trial	Efficiency			Success
	robot	flock	total	
1	14.22	11.64	25.86	1.87
2	25.16	13.59	38.75	1.96
3	12.81	11.17	23.98	1.74
4	13.3	11.7	25	1.79
5	16.13	14.33	30.46	2.09
6	9.45	9.27	18.72	1.45
7	13.89	11.16	25.05	1.85
8	14.87	13.43	28.3	1.91
9	12.6	10.57	23.17	1.70
10	12.8	10.3	23.1	1.68
11	14.37	12.05	26.42	1.85
12	12.51	10.05	22.56	1.69
total	172.11	139.26	311.37	21.58
average	14.34	11.61	25.95	1.80
variance	14.26	2.35	25.01	0.03
stdev	3.78	1.54	5.00	0.16

Table 5.1: Summary of results for Method 2 simulation trials. All results are measured in metres.

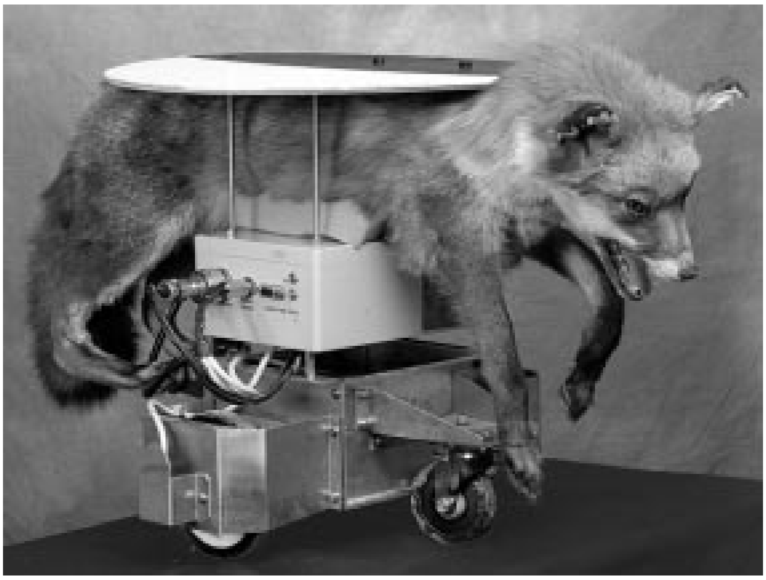


Figure 5.15: Rover with Jane Henderson's stuffed fox mounted, as used in the first Method 2 real world trials.

position **RS**, robot finish position **RF**, flock start **FS** and flock finish **FF**. The plots have been rotated so that the goal position is always at the bottom of the arena at (0,-3.5).

Also given are plots of the flock distance-to-goal (left) and the distance traveled by the robot and the flock (right). These data are used to calculate final success and efficiency scores for each trial as described in the previous chapter. These scores are presented in Table 5.1 along with their averages and variances.

The average success score over 12 trials for Method 2 in simulation was 1.8m. The average efficiency score over 12 trials was 25.95m, of which 11.61m was flock movement. These trials were identical to those performed for Method 1, so the results can be compared; Method 2 gives a 22% improvement in success and a 23% improvement in efficiency over Method 1 in simulation.

5.4 Real world trials

Method 2 was then transferred on to the physical robot, to be tested in the real world.

In order to fit in with Jane Henderson's experimental program, the physical appearance of the vehicle was changed for these trials. A stuffed fox was mounted between Rover's cover and chassis. This enabled Jane to examine the responses of the ducks to the presence of the fox. Figure 5.4 shows the modified vehicle. In order to maintain the front of the fox facing the flock, the steering algorithm described in chapter 3 was modified to allow forwards movement only. This could slow Rover's response to the flock-controller's goal vector, as it may have to turn 180^o instead of the maximum 90^o required when allowed to move in either direction.

For comparison with Method 1 trials, it would have been preferable to keep the appearance and control of the vehicle the same, but the ducks were an expensive, limited resource and there was pressure to save time and costs by combining Jane and the author's experiments. As the results below will show, this combination was not very successful, so further experiments with the original configuration were necessary.

5.4.1 Procedure

With the exception of the above modifications to the vehicle and controller, the experimental procedure was identical to that of the Method 1 real world trials described in the previous chapter. Identical data were recorded from each trial and are presented below.

The same three flocks of twelve ducks each were used as for the Method 1 trials. They were still subject to other behaviour experiments, so had received identical husbandry and handling.

5.4.2 Results

Results are presented for each trial in Figures 5.16 to 5.24. A plot of the paths of the robot and flock centre through the arena is given (top). The path plots are labelled with the robot start position **RS**, robot finish position **RF**, flock start **FS** and flock finish **FF**. The plots have been rotated so that the goal position is always at the bottom of the arena at (0,-3.5).

Also given are plots of the flock distance-to-goal (left) and the distance traveled by the robot and the flock (right). These data are used to calculate final success and efficiency scores for each trial as described in the previous chapter. These scores are presented in Table 5.2 along with their averages and variances.

The average score over the 5 admitted trials was 1.85m. The average score over the 5 admitted trials was 46.48m, of which 20.21m was flock movement.

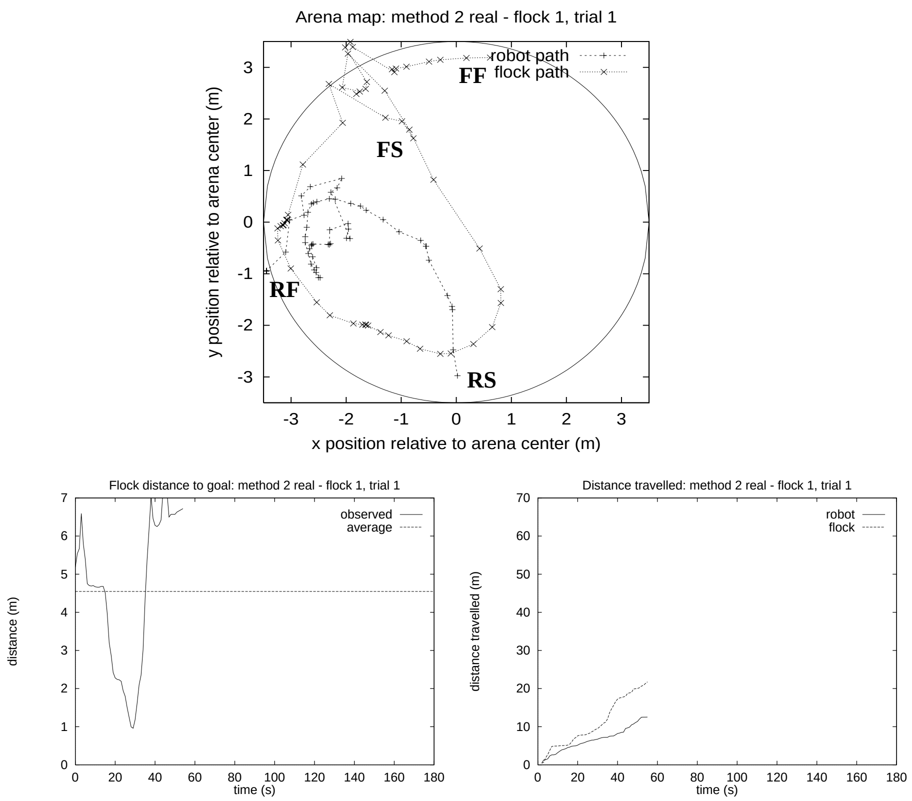


Figure 5.16: Method 2 real-world results - trial 1.

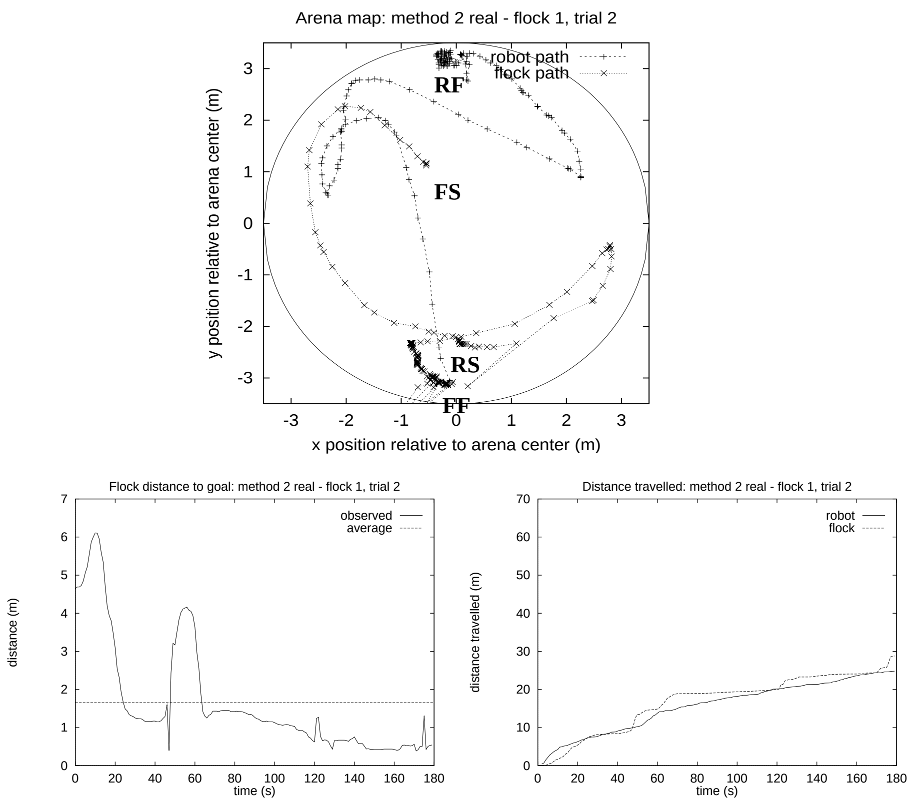


Figure 5.17: Method 2 real-world results - trial 2.

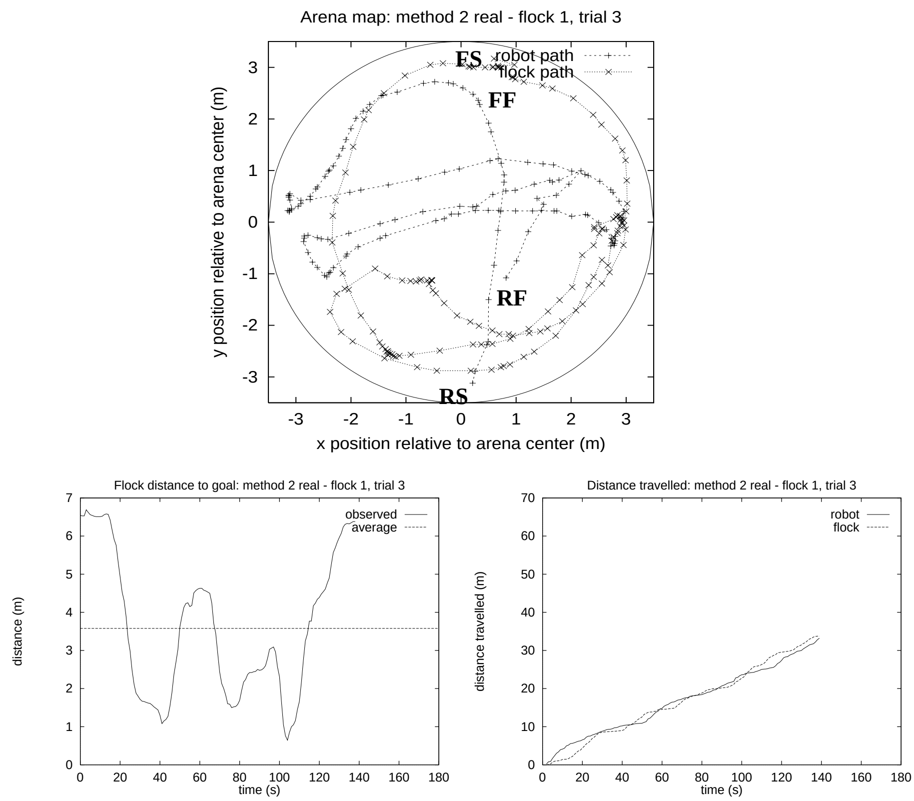


Figure 5.18: Method 2 real-world results - trial 3.

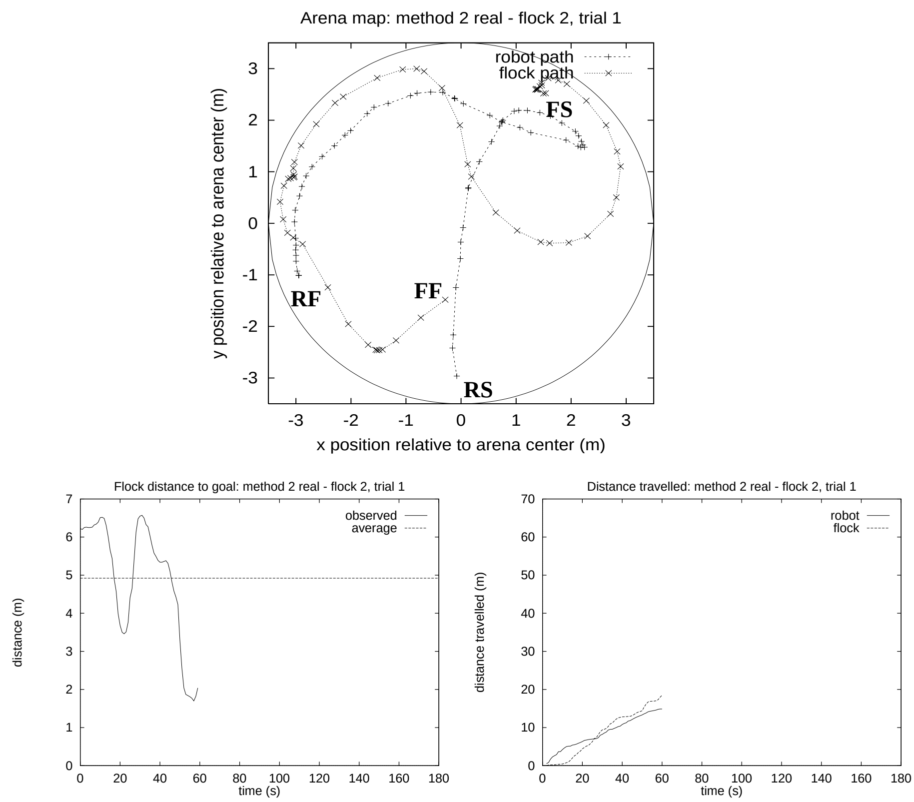


Figure 5.19: Method 2 real-world results - trial 4.

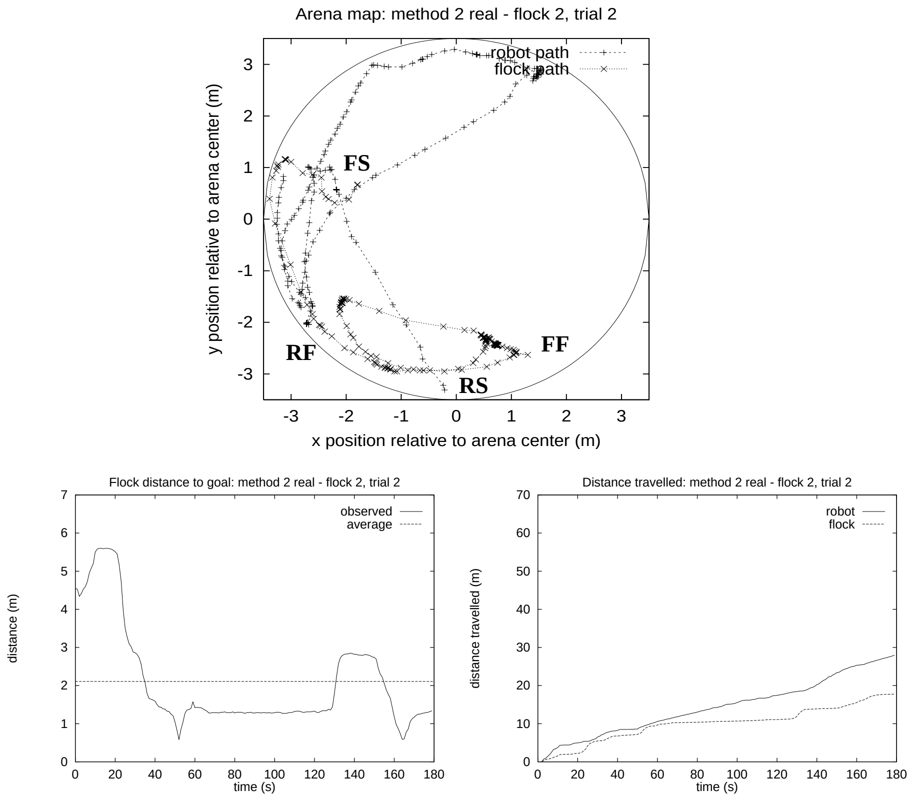


Figure 5.20: Method 2 real-world results - trial 5.

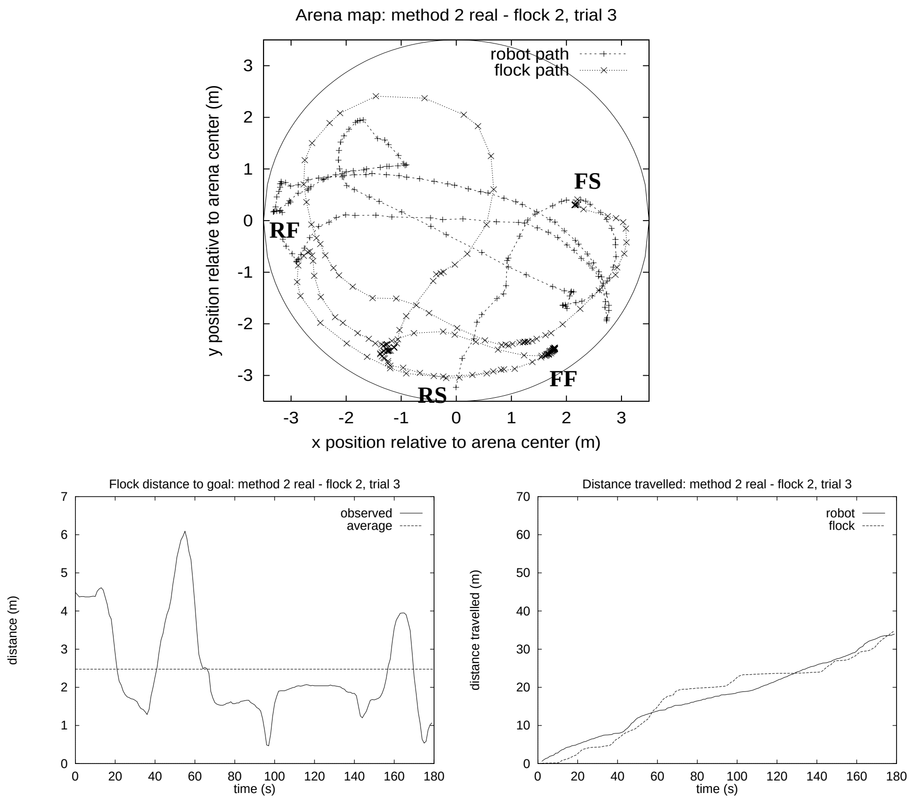


Figure 5.21: Method 2 real-world results - trial 6.

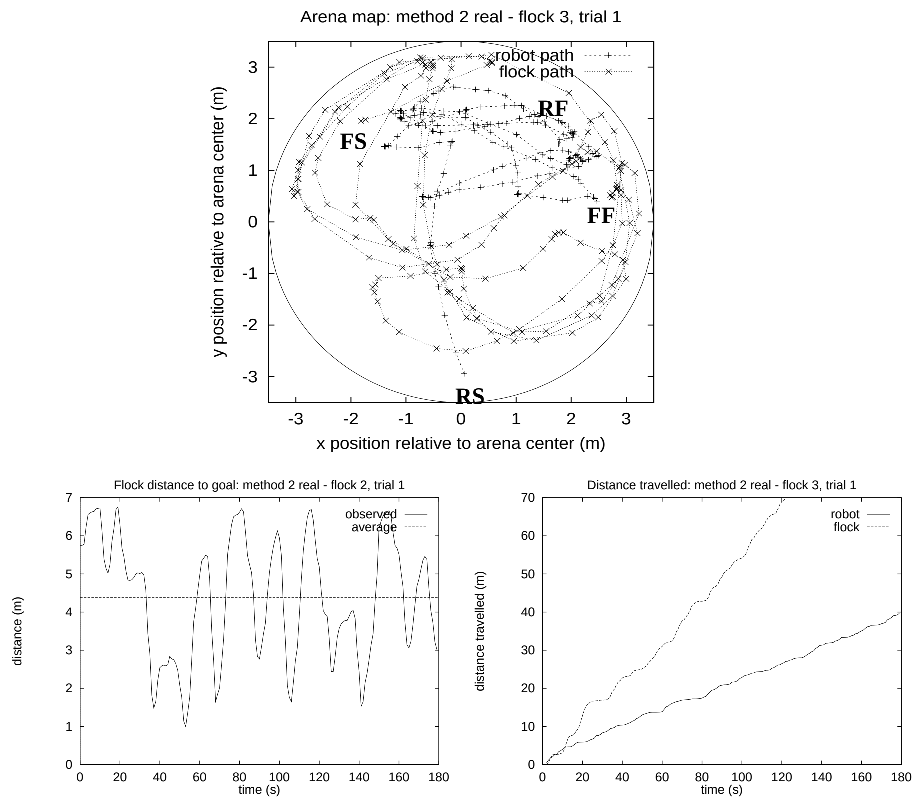


Figure 5.22: Method 2 real-world results - trial 7.

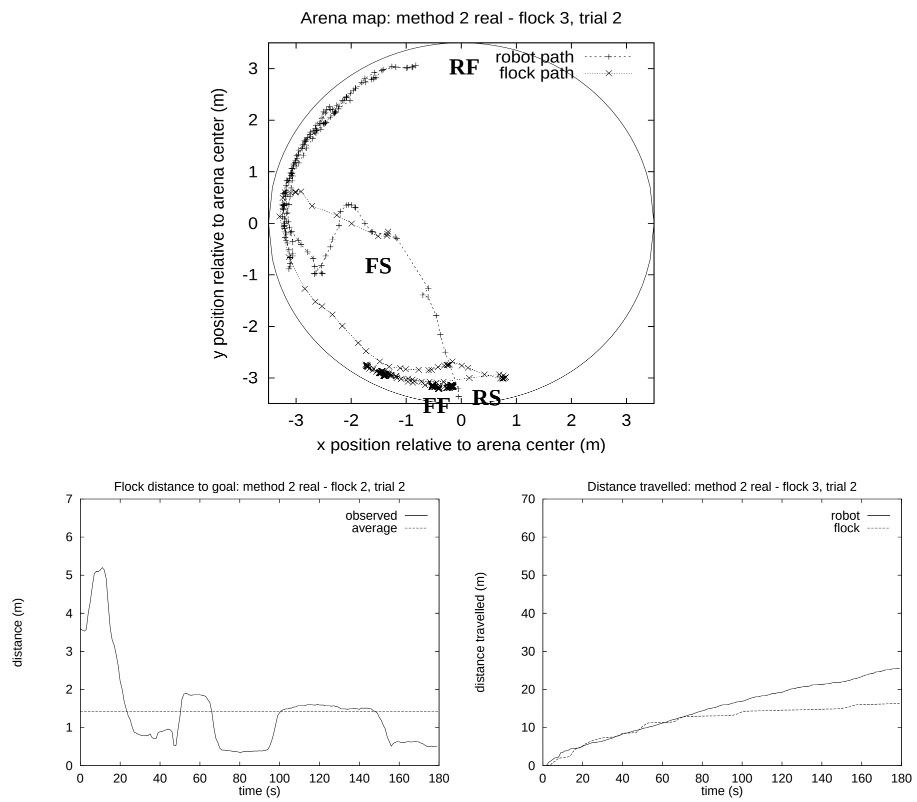


Figure 5.23: Method 2 real-world results - trial 8.

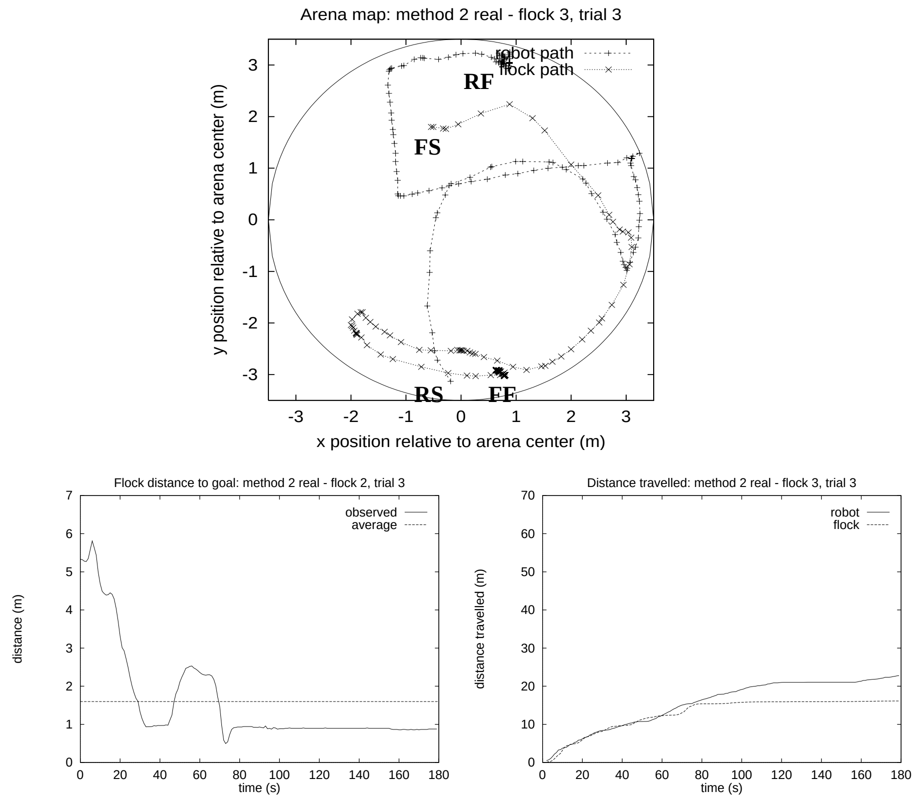


Figure 5.24: Method 2 real-world results - trial 9.

Trial	Efficiency			Success
	robot	flock	total	
1	-	-	-	-
2	22.77	16.12	38.89	1.65
3	-	-	-	-
4	-	-	-	-
5	27.94	17.79	45.73	2.11
6	33.96	34.67	68.63	2.48
7	-	-	-	-
8	25.57	16.34	41.91	1.42
9	22.77	16.12	38.89	1.60
total	133.01	101.04	234.05	9.25
average	26.60	20.21	46.81	1.85
variance	21.59	65.85	156.69	0.19
stdev	4.65	8.11	12.52	0.43

Table 5.2: Summary of results for Method 2 real-world trials. All results are measured in metres.

5.4.3 Example real world trials

Trial 2, shown in Figure 5.17, worked well and showed the greatest similarity to the characteristic behaviour seen in the simulation. The robot starts at the goal and moves towards the flock, which starts just left of the arena centre. As the flock reaches the arena wall, the robot is above it with respect to the goal, so the flock moves down towards the goal. As the flock approaches the goal the robot backs off towards the far side of the arena. The flock overshoots the goal once, but returns as the robot backs retreats, coming again to the left of the goal, then finally settling close to the goal, with the robot at the far side of the arena.

Trial 9, shown in Figure 5.17, also looks similar to the simulations, but shows the effect of the increased threat felt by the ducks. The robot initially approaches the flock which starts at the top middle of the arena. The flock moves down the wall towards the goal, 'pushed' by the robot. But they overshoot the goal by a large amount because the robot is too close and doesn't back away in time. The robot moves quickly across the arena to again push the flock towards the goal. As they approach the goal again, the robot backs away as the flock-attraction becomes small and the goal-repulsion dominates. The flock settles near the goal, with the robot on the far side of the arena. As in the original Method 1 pilot trial, the real robot system exhibits a behaviour in response to the ducks that was not seen in the simulation, but results in successful flock-control.

5.4.4 Discussion

These trials were rather unsuccessful. A combination of vision system failures, external disturbance and the fox mounted on the robot gave poor results. Trials 1, 3, & 4 were halted because the flock tracker lost the ducks and could not be manually reset. After around 60 seconds of trial 7 a loud noise outside the workshop alarmed the ducks, causing them to rush around the arena in panic. The trials could not be repeated because the ducks were subject to other behaviour experiments, which could have been confounded by unequal exposure to the robot and arena.

Of the remaining five trials, three succeeded in bringing the ducks to the goal. It was observed that the ducks avoided the fox-robot much more strongly than the plain-cover robot in the Method 1 trials. The ducks seemed far more 'flighty', vocalizing and flapping their wings through parts of the trial. This was not witnessed in the Method 1 trials. It seems that the extra threat perceived by the ducks from this robot made them harder to control.

In addition, the dynamics of the robot were different; the weight of the fox increased the inertia, etc, making the real vehicle behave even less like the idealized simulation than before. The constraint of having to move in one direction only meant that the robot turned much more slowly than in its standard configuration. These factors will have contributed to the poor performance.

The greatly increased avoidance of the new-look robot by the ducks was not featured in the model used to develop the new method. However, in three out of 5 complete real world trials, Method 2 succeeded in gathering the flock. These results, while apparently poorer than those for Method 1 are somewhat reassuring as they demonstrate that the method can cope with a very different flock to the one it's parameters were adjusted for. However, these differences in the experimental set up and the poor results obtained make it impossible to directly compare the performance of Methods 1 and 2.

5.5 Further trials

Due to the vision system failures and noise disturbance during the Method 2 real-world trials, seven more trials were performed to better assess the algorithm's performance.

5.5.1 Procedure

This time, for a fairer comparison of Method 2 with Method 1's real world performance, the robot vehicle was reconfigured to be the same as for the Method 1 experiments, with the plain short cover. The steering mechanism was reset to allow movement in both directions, as for the Method 1 experiments.

Only one flock of twelve ducks was available for these trials, of a different breed to the previous real world trials. This flock had not been used for other experimental work. The use of a different breed and a different number of birds is likely to produce a slightly different overall flock behaviour, but this variation is well within the range in which the robot is desired to work. The original breed was a meat-producing variety which had trouble moving around comfortably once they reached 8 weeks old and were due for slaughter, as they had gained a great deal of weight. An egg-producing breed was chosen so that the individuals would stay fit enough to be useful for several experiments.

Trial	Efficiency			Success
	robot	flock	total	
10	15.84	17.96	33.80	1.12
11	18.73	14.25	32.98	1.88
12	16.71	14.20	30.91	1.82
13	15.50	30.18	45.68	1.71
14	20.36	41.65	62.01	1.86
15	17.71	16.83	34.54	1.85
16	18.29	9.83	28.12	1.46
total	256.15	245.94	502.09	11.70
average	21.35	20.49	38.29	1.67
variance	2.95	125.54	139.59	0.08
stdev	1.72	11.20	11.81	0.28

Table 5.3: Summary of results for further Method 2 real-world trials. All results are measured in metres.

Henderson [Henderson, 1999] describes the rearing conditions for these birds.

The experimental procedure was identical in all other respects to that used for previous trials, with the same data logged.

5.5.2 Results

Results are presented as before for each trial, numbered from 10 to 16 in Figures 5.26 to 5.32. The success and efficiency scores are presented in Table 5.3 along with their averages and variances. The average success score over the 7 trials was 1.67m. The average efficiency score over the 7 trials was 41.84m, of which 20.49m was flock movement.

By the success <2m criterion, Method 2 worked in 100% (7/7) of these further trials.

5.5.3 Example further real world trial

Figure 5.25 shows a sequence of views from the overhead camera during trial 13 (Figure 5.29). The duck and robot behaviour observed in this trial closely resembles the simulation. The trial starts with the robot near the goal and the ducks right and slightly above the arena center (1). The robot moves towards the ducks (2) which closely aggregate and move away from the robot to the arena wall (3). The robot moves behind the flock and pushes them towards the goal (5). As the flock gets close to the goal and Method 2's flock-attraction is reduced, the goal-repulsion dominates and the robot increase its distance from the flock, reducing the stimulus on them to move (6). With the flock very close to the goal, the flock-attraction acting on the robot is near zero, so the robot

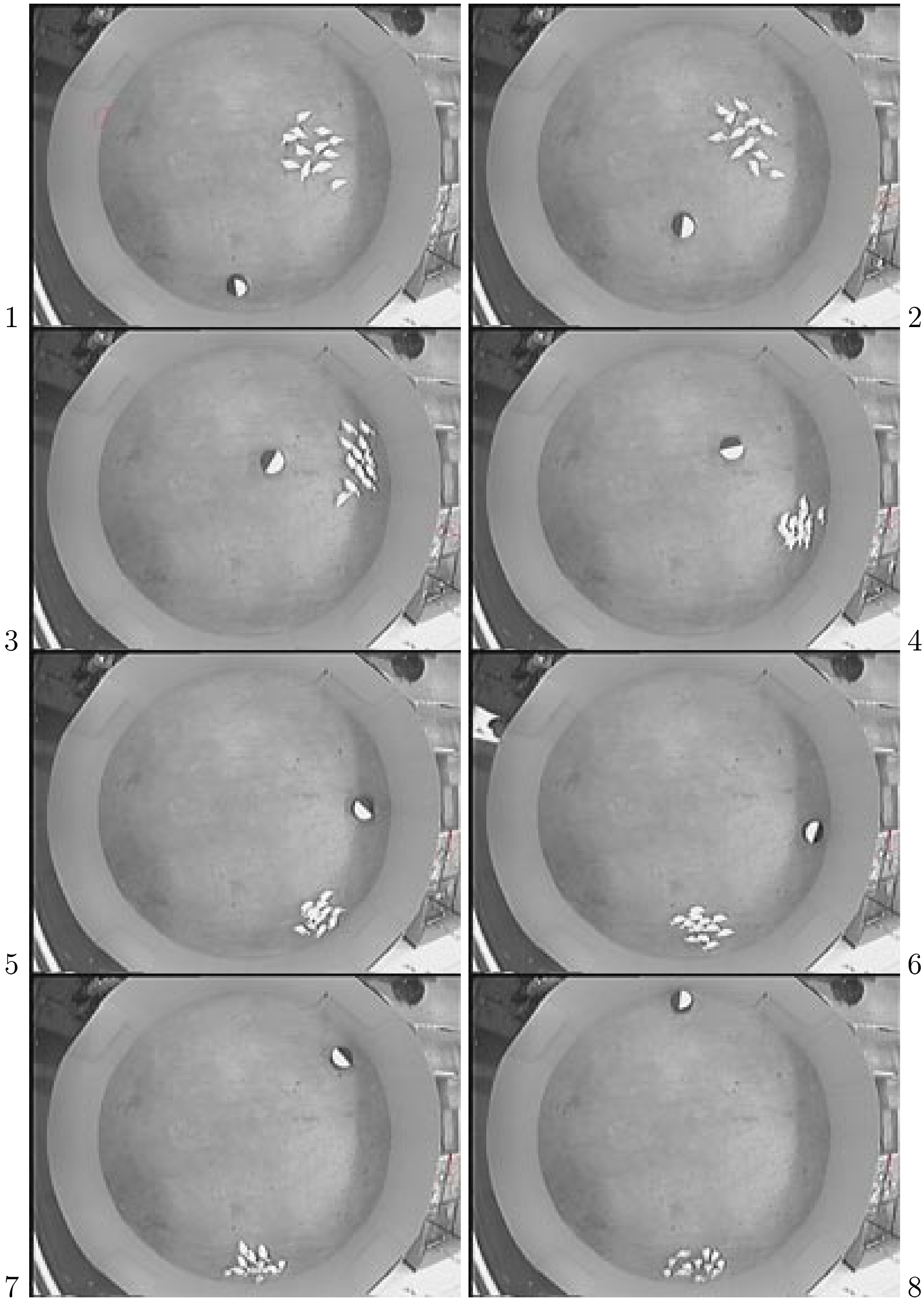


Figure 5.25: Sequence of images from the overhead camera during an experiment, showing successful behaviour. The goal position is at the bottom of the picture.

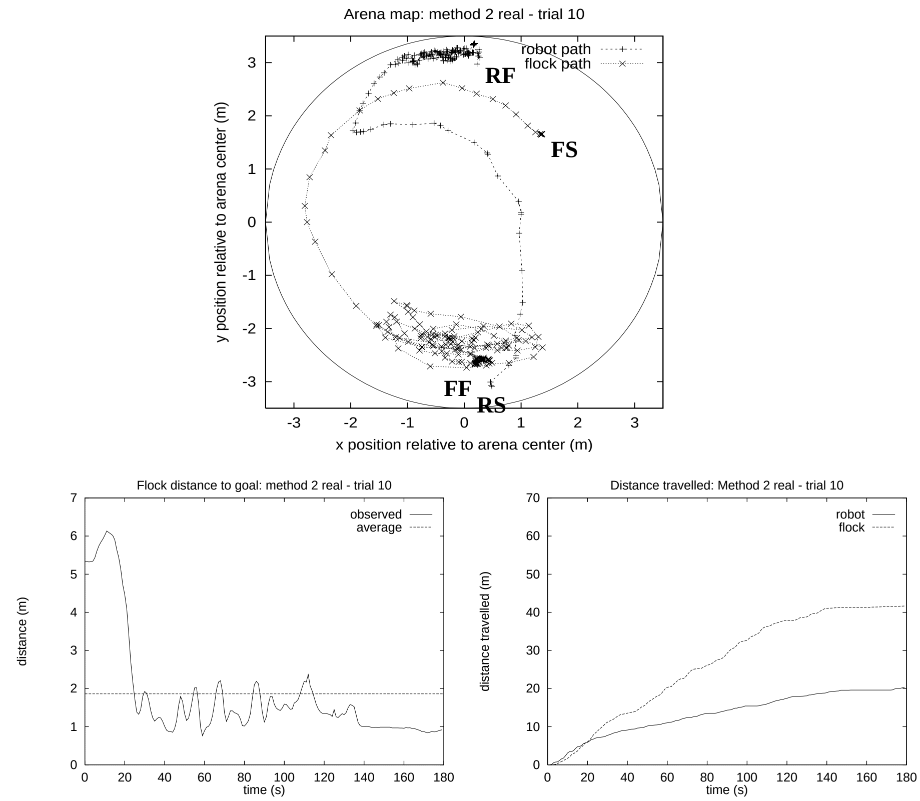


Figure 5.26: Further Method 2 real-world results - trial 10

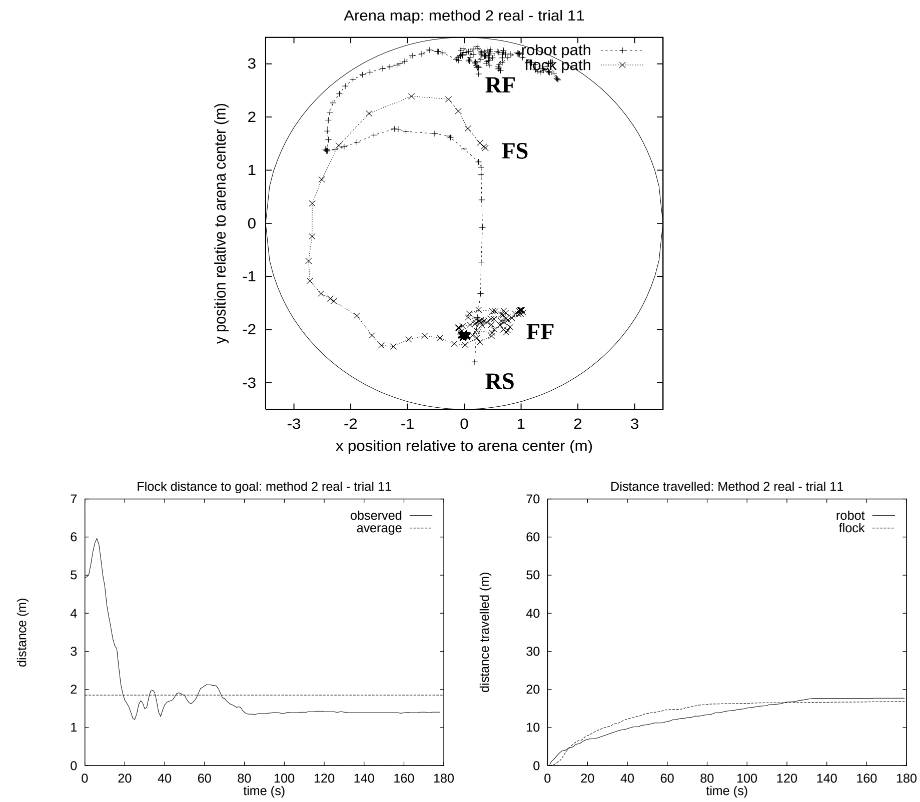


Figure 5.27: Further Method 2 real-world results - trial 11

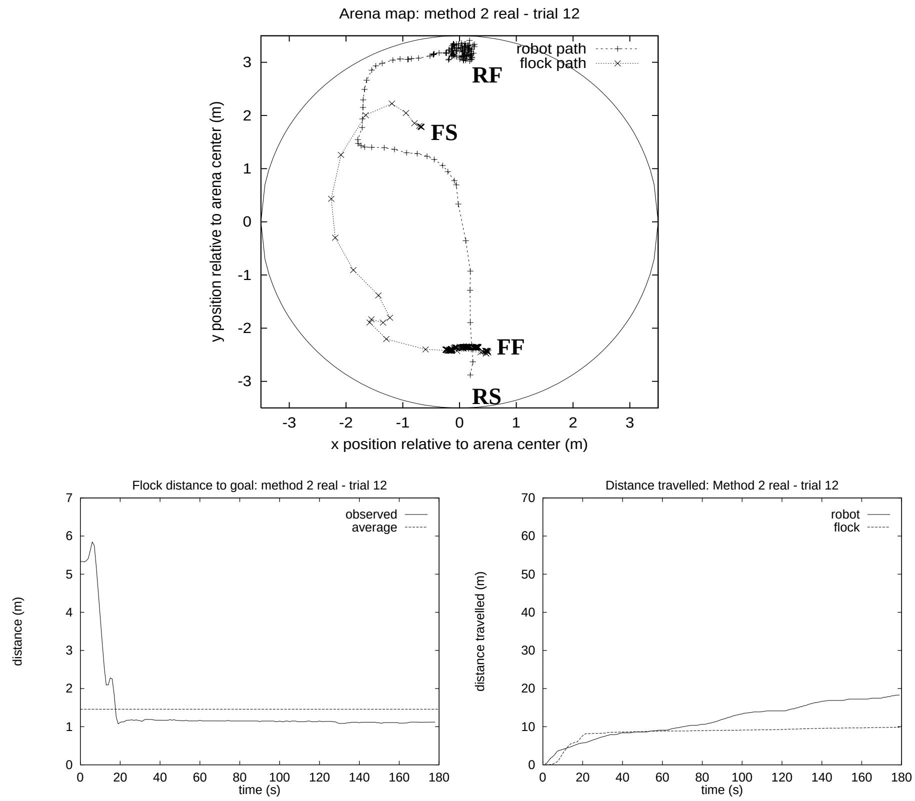


Figure 5.28: Further Method 2 real-world results - trial 12

moves away from the goal (7) and spends the rest of the trial at the far side of the arena with the ducks at the goal.

The plot of the robot and duck paths for this trial look very similar to those produced by this method in the simulator. Compare Figure 5.29 with Figure 5.4. A good success score of 1.71m is achieved, so this trial is considered to have worked.

5.5.4 Discussion

The path plots of all these trials all have a similar shape, indicating a much more consistent system performance in this set of trials. The oscillation of the flock about the goal, characteristic of Method 1, is not observed, and the efficiency scores are therefore much better.

The path plots also indicate that this flock of ducks kept a greater distance from the robot than the flocks in the Method 1 trails. Figure 5.26 shows that the robot switched from moving towards the ducks to moving away from the goal very high in the arena. For this to happen, the ducks must have been close to the goal; around 4m from the robot.

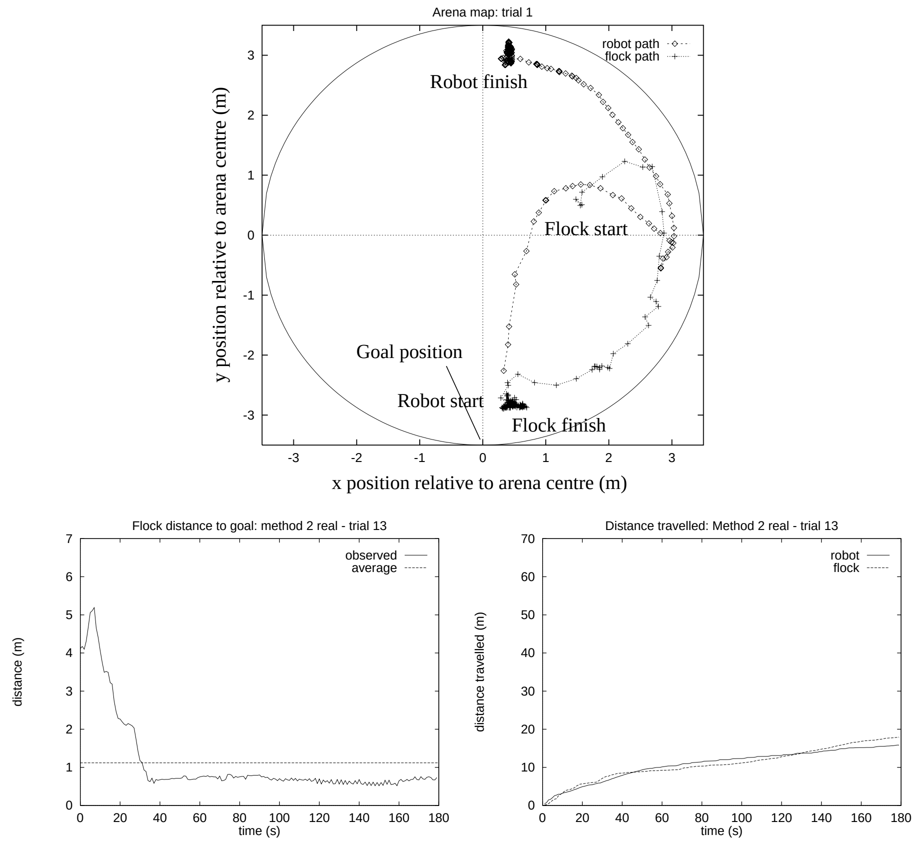


Figure 5.29: Further Method 2 real-world results - trial 13

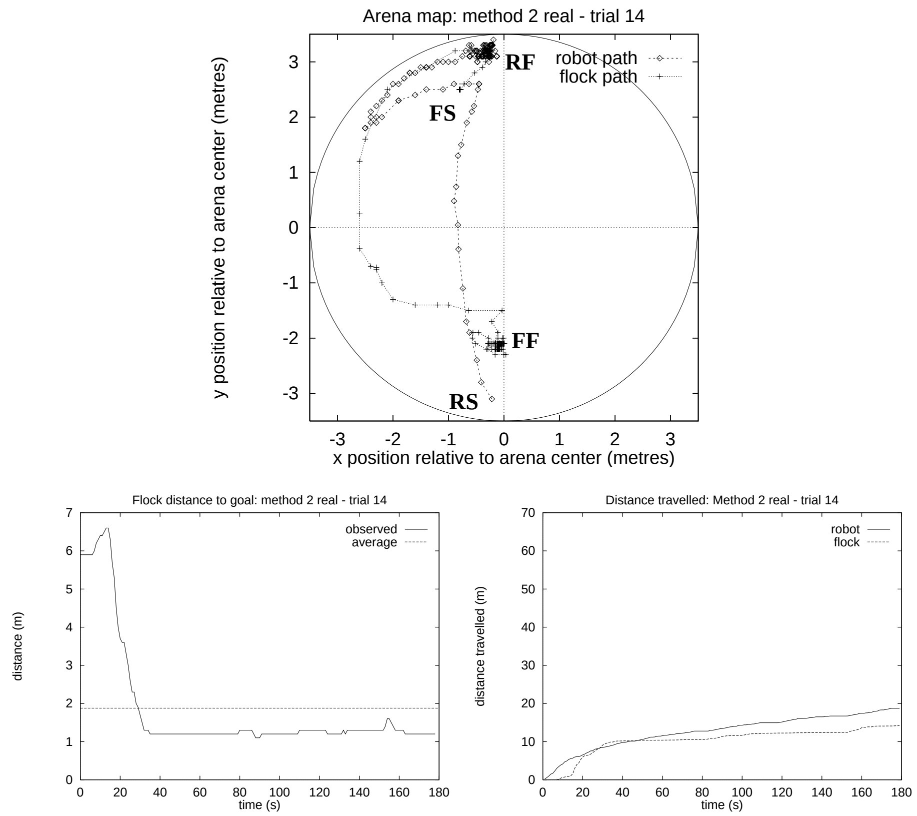


Figure 5.30: Further Method 2 real-world results - trial 14

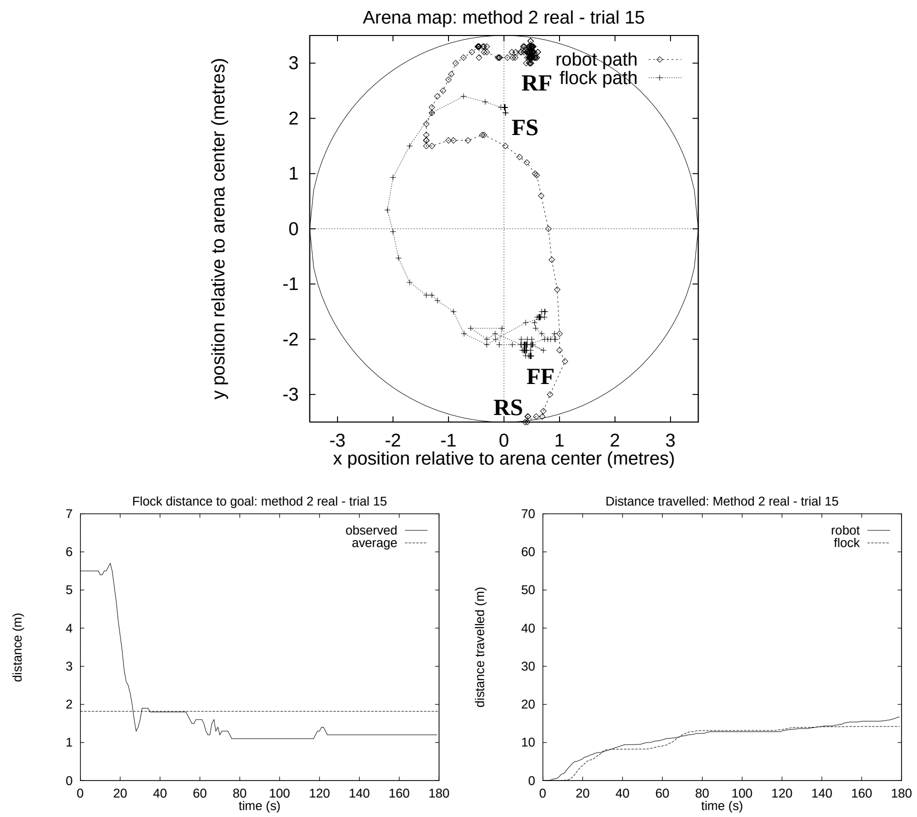


Figure 5.31: Further Method 2 real-world results - trial 15

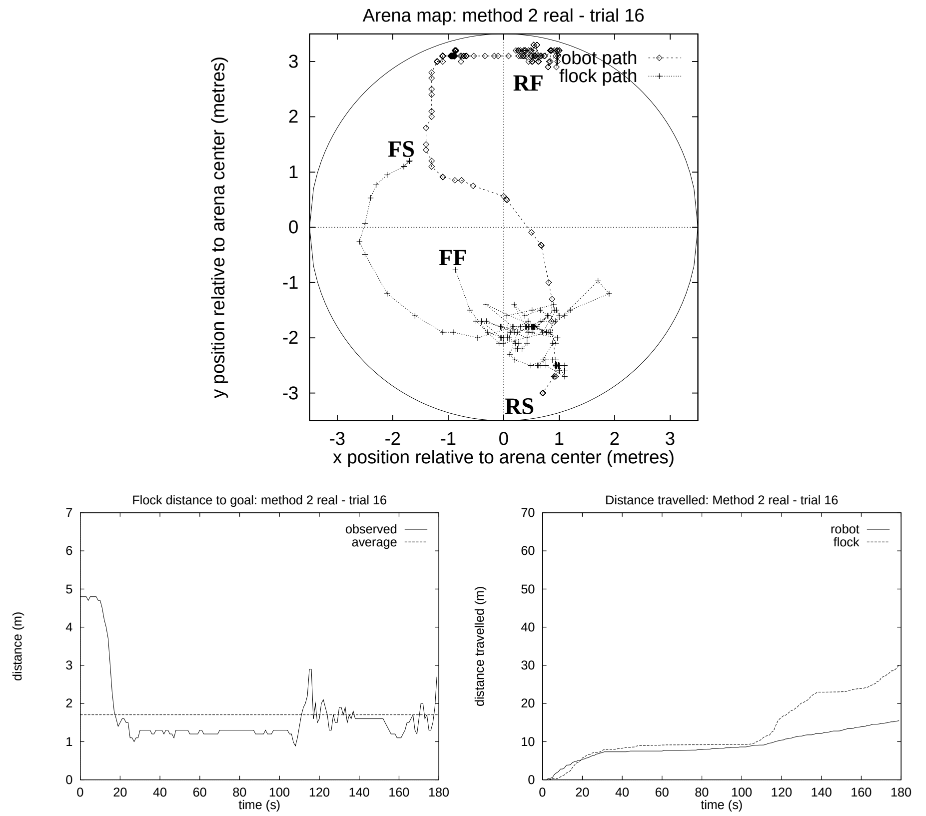


Figure 5.32: Further Method 2 real-world results - trial 16

The previous set of trials (trials 1-9) seemed to be less successful because the flock was too fearful of the fox-robot, and were flighty and hard to control. These trials (10-16) show that while these ducks avoid the robot by a greater distance than the previous flocks in the Method 1 trials, they are still controllable. It seems likely that the plain-cover robot causes less stress to the flock than the fox-robot, and this is what makes them more controllable. However, it is recognized that the differences between the two sets of trials (different duck breed, different number of birds) mean that this was not a definitive experiment. It would be useful and interesting to clarify this point with another set of experiments, but lack of time precluded this.

5.6 Conclusions

(1) Adapting the original flock control algorithm in the style of a proportional controller solves the oscillation problem and gives an improved performance. Method 2 is more stable and effective than Method 1 by means of its proportional design, and is therefore a more appropriate solution.

(2) While a full control-theory analysis of an animal-interactive system would not be viable, an engineering principle such as proportional control can still be usefully applied.

(3) The robot with a plain cover proved more successful than the robot with a fox mounted on it. This suggests that the less threatening appearance of the robot makes flock control easier by reducing the stress on the animals.

Chapter 6

Discussion

This chapter further examines the results from the previous experimental chapters, comparing them with a pair of control experiments.

The abilities and limitations of the algorithms and experiments are discussed, and some questions and criticisms encountered in the course of the work are addressed.

6.1 Comparing the simulated and real results

Figures 6.1 and 6.3 show plots of the final success (x axis) and efficiency (y axis) scores of the simulated and real experiments respectively. Recall that a lower score on either axis indicates better performance. Boxes are drawn around the extremes from each trial to indicate the range of values observed.

The average efficiency/success score from each experiment is presented as a vector in Figures 6.2 and 6.4. The length of the vector gives a simple 'goodness' indicator by straightforwardly combining the success and efficiency scores. The y axis scale (success=0 $\rightarrow$ 7m) was chosen as the largest possible range of success scores, while the x axis (efficiency=0 $\rightarrow$ 110m) is just larger than the largest efficiency score observed. By this metric Method 2 was better than Method 1 both in simulation and in the real world; agreeing with the qualitative assessment given in the previous chapter.

The boxes in Figure 6.2 and 6.4 indicate $\pm$ 1 standard deviation from the mean in both axes. This is provided as a rough indication of the deviation only, as the assumption of a normal distribution about the mean does not necessarily hold, and again the sample size is very small.

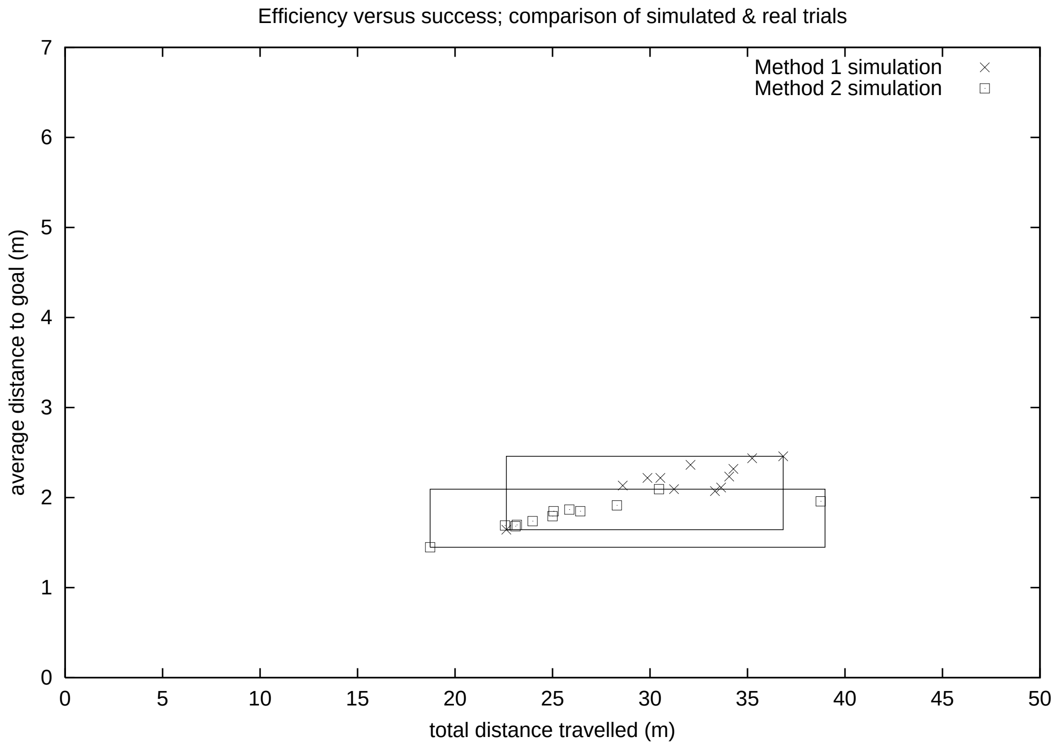


Figure 6.1: Distribution of success versus efficiency scores in the simulation trials. Boxes indicate range of results.

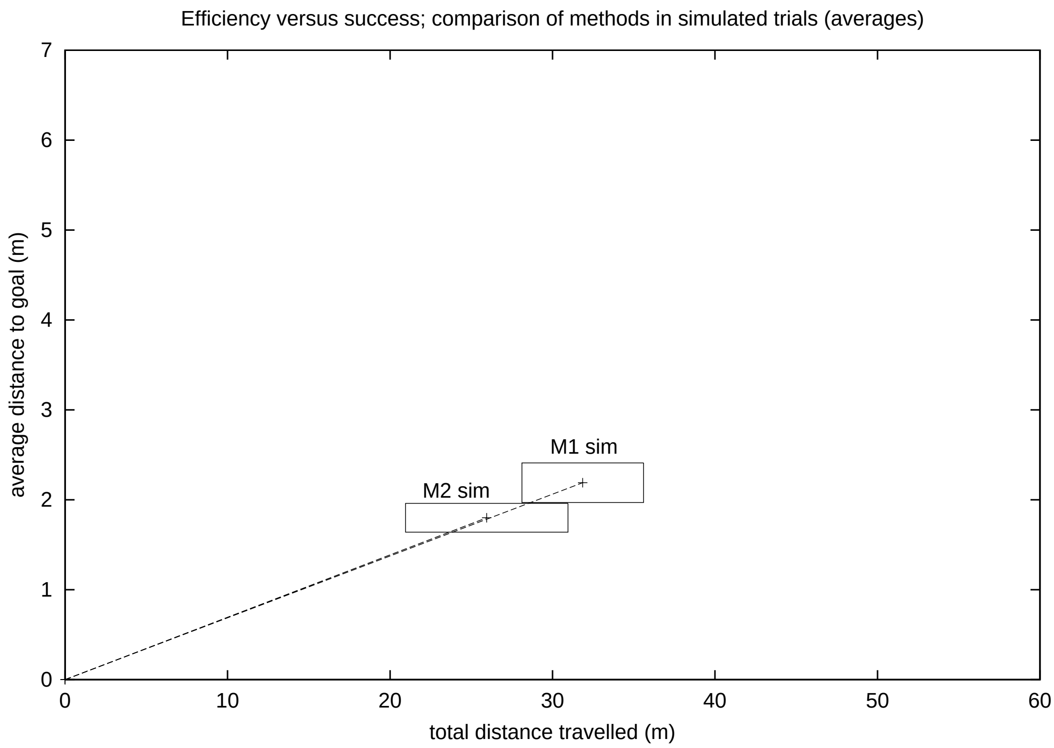


Figure 6.2: Mean success/efficiency scores for the simulation trials. Boxes indicate standard deviation from the mean.

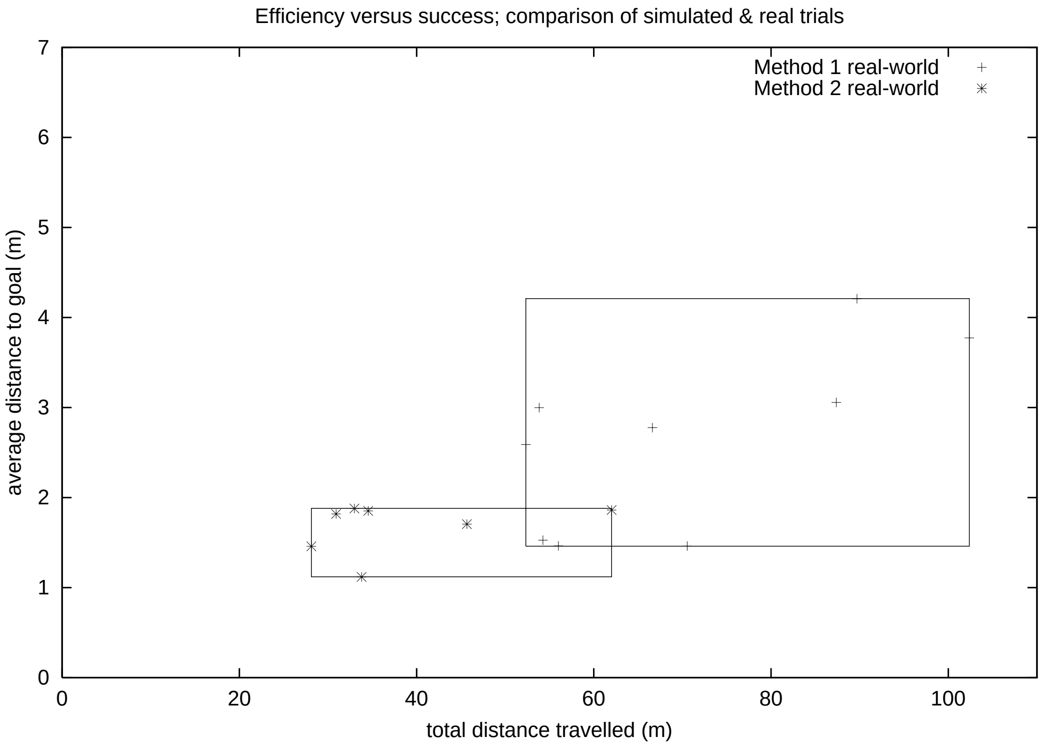


Figure 6.3: Distribution of success versus efficiency scores in the simulation trials. Boxes indicate range of results.

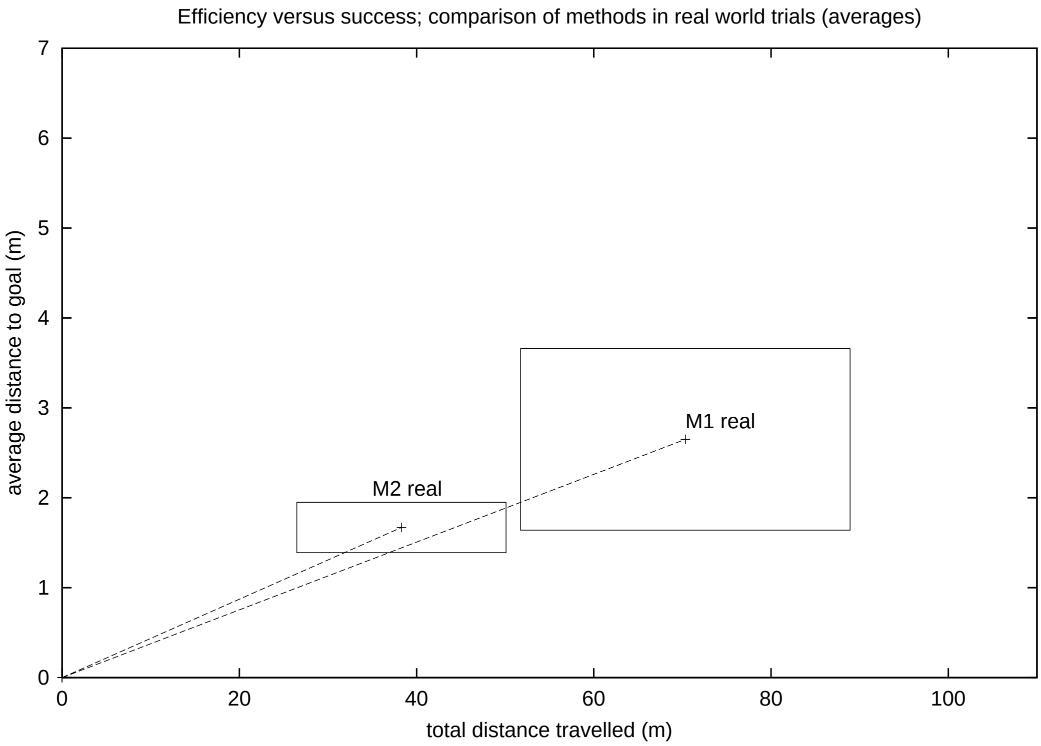


Figure 6.4: Mean success/efficiency scores for the simulation trials. Boxes indicate standard deviation from the mean.

6.1.1 Statistical analysis

The success scores for Methods 1 and 2 are compared to each other and to a sample of random scores to determine whether the results are significantly different.

The hypotheses suggested by the qualitative analysis above are that in both the simulation and the real world:

1. Method 1 performs better, i.e. gathers the ducks closer to the goal, than if the flock was positioned randomly in the arena.

2. Method 2 performs better than Method 1.

As a lower success score is better, the combined hypothesis is that of the recorded success scores, Method 2 $<$ Method 1 $<$ random. Six comparisons are done to test this hypothesis.

Method 1 vs. random - simulation

The null hypothesis (H_0) is that in simulation Method 1 shows results that do not differ significantly from those produced by a set of flocks randomly positioned in the arena.

The alternative hypothesis (H_1) is that the Method 1 (m1) controller performs better than random, ie. a sample from the population of m1 scores is likely to have a lower success score than a sample from the random population.

The Mann-Whitney U test is chosen as an appropriate test. This test is used to determine the probability that two independent groups are drawn from the same population and is appropriate for small samples. While the data analysed here is measured on a ratio scale (evenly-spaced success scores from 0 [perfect] to infinity [bad]) and the U test only requires an ordinal scale, the U test avoids the assumption of normal distribution and equal variance between the populations that is required by the t test [Siegel, 1956]. The same test will be used in all the following analyses.

Since H_1 predicts the direction of the hypothesised difference, the region of rejection is one-tailed. Any computed value for U that has a probability of occurring under H_0 equal to or less than $\alpha = 0.05$ will reject H_0.

The data sets for this comparison are presented in Table 6.1 (top left). The samples of type m1s are from the Method 1 simulation trials. The sample of type rnd are the distance-to-goal

Method 1 sim vs. random

type	success	rank
m1s	1.64	1
m1s	2.07	2
m1s	2.09	3
m1s	2.11	4
m1s	2.13	5
m1s	2.22	7
m1s	2.22	8
m1s	2.23	9
m1s	2.32	10
m1s	2.36	11
m1s	2.44	12
m1s	2.46	13
rnd	2.176	6
rnd	2.658	14
rnd	2.926	15
rnd	2.954	16
rnd	3.212	17
rnd	3.227	18
rnd	3.602	19
rnd	4.059	20
rnd	4.107	21
rnd	4.384	22
rnd	5.187	23
rnd	6.268	24

type	rank total	U/U'	U value
m1s	85	U'	137
rnd	215	U	7

Method 2 sim vs. random

type	success	rank
m2s	1.45	1
m2s	1.68	2
m2s	1.69	3
m2s	1.7	4
m2s	1.74	5
m2s	1.79	6
m2s	1.85	7
m2s	1.85	8
m2s	1.87	9
m2s	1.91	10
m2s	1.96	11
m2s	2.09	12
rnd	2.176	13
rnd	2.658	14
rnd	2.926	15
rnd	2.954	16
rnd	3.212	17
rnd	3.227	18
rnd	3.602	19
rnd	4.059	20
rnd	4.107	21
rnd	4.384	22
rnd	5.187	23
rnd	6.268	24

type	rank total	U/U'	U value
m2s	78	U'	144
rnd	222	U	0

Method 1 real vs. random

type	success	rank
m1r	1.46	2
m1r	1.46	3
m1r	1.53	4
m1r	2.59	5
m1r	2.78	6
m1r	3.00	7
m1r	3.06	8
m1r	3.77	16
m1r	4.21	17
rnd	1.359	1
rnd	3.301	9
rnd	3.336	10
rnd	3.406	11
rnd	3.572	12
rnd	3.643	13
rnd	3.696	14
rnd	3.756	15
rnd	5.031	18

type	rank total	U/U'	U value
m1r	68	U'	58
rnd	103	U	23

Method 2 real vs. random

type	success	rank
m2r	1.12	1
m2r	1.46	2
m2r	1.71	4
m2r	1.82	6
m2r	1.85	7
m2r	1.86	8
m2r	1.88	9
rnd	1.599	3
rnd	1.765	5
rnd	2.652	10
rnd	3.217	11
rnd	4.347	12
rnd	4.767	13
rnd	4.881	14

type	rank total	U/U'	U value
m2r	37	U'	40
rnd	68	U	9

Table 6.1: Table showing the data and intermediate results used to calculate the Mann-Whitney U tests for experimental trials (Table 1 of 2).

Method 1 sim vs. Method 2 sim

type	success	rank
m1s	1.64	2
m1s	2.07	13
m1s	2.09	14
m1s	2.11	16
m1s	2.13	17
m1s	2.22	18
m1s	2.22	19
m1s	2.23	20
m1s	2.32	21
m1s	2.36	22
m1s	2.44	23
m1s	2.46	24
m2s	1.45	1
m2s	1.68	3
m2s	1.69	4
m2s	1.7	5
m2s	1.74	6
m2s	1.79	7
m2s	1.85	8
m2s	1.85	9
m2s	1.87	10
m2s	1.91	11
m2s	1.96	12
m2s	2.09	15

type	rank total	U/U'	U value
m1s	209	U	13
m2s	91	U'	131

Method 1 real vs. Method 2 real

type	success	rank
m1r	1.46	3
m1r	1.46	4
m1r	1.53	5
m1r	2.59	11
m1r	2.78	12
m1r	3	13
m1r	3.06	14
m1r	3.77	15
m1r	4.21	16
m2r	1.12	1
m2r	1.46	2
m2r	1.71	6
m2r	1.82	7
m2r	1.85	8
m2r	1.86	9
m2r	1.88	10

type	rank total	U/U'	U value
m1r	93	U	15
m2r	43	U'	48

Table 6.2: Table showing the data and intermediate results used to calculate the Mann-Whitney U tests for experimental trials (Table 2 of 2).

scores found for randomly generated points within the arena. The table shows the ranking of the scores and the total of the ranks for both sample types, used to determine the U score. The sample sizes are $n_1 = 12$ (m1s) and $n_2 = 12$ (rnd).

Following the algorithm for the U test [Siegel, 1956, pp116-127] it is found that the U value for these data sets is 7. From the Mann-Whitney U test tables it is found that the critical U for a one-tailed test at $\alpha = 0.05$ is 42. Any value of U less than or equal to 42 indicates that the probability of the two data sets being drawn from the same population (H_0) is less than 0.05. H_0 is therefore rejected in favour of H_1. It is shown that in simulation Method 1 shows a significantly different average flock-to-goal distance to a random set of flock positions, and that the average is smaller for Method 1.

Method 2 vs. random - simulation

The null hypothesis (H_0) is that in simulation Method 2 shows results that do not differ significantly from those produced by a set of flocks randomly positioned in the arena.

The alternative hypothesis (H_1) is that the Method 2 (m2) controller performs better than random, ie. a sample from the population of m2 scores is likely to have a lower success score than a sample from the random population.

The data sets for this comparison are presented in Table 6.1 (top right). The samples of type *m2s* are from the Method 2 simulation trials. The sample of type *rnd* are the distance-to-goal scores found for randomly generated points within the arena. The sample sizes are $n_1 = 12$ (m2s) and $n_2 = 12$ (rnd).

It is found that the U value for these data sets is 0. From the Mann-Whitney U test tables it is found that the critical U for a one-tailed test at $\alpha = 0.05$ is 42. H_0 is therefore rejected in favour of H_1. It is shown that in simulation Method 2 shows a significantly different average flock-to-goal distance to a random set of flock positions, and that the average is smaller for Method 2.

Method 1 vs. random - real world

The null hypothesis (H_0) is that in real world trials Method 1 shows results that do not differ significantly from those produced by a set of flocks randomly positioned in the arena.

The alternative hypothesis (H_1) is that the Method 1 (m2) controller performs better than random, ie. a sample from the population of m1 scores is likely to have a lower success score than a sample from the random population.

The data sets for this comparison are presented in Table 6.1 (bottom left). The samples of type $m1s$ are from the Method 1 real-world trials. The sample of type rnd are the distance-to-goal scores found for randomly generated points within the arena. The sample sizes are $n_1 = 9$ (m1s) and $n_2 = 9$ (rnd).

It is found that the U value for these data sets is 23. From the Mann-Whitney U test tables it is found that the critical U for a one-tailed test at $\alpha = 0.05$ is 21. H_0 cannot be rejected on the basis of these results. The results of Method 1's trials in the real world were not shown to be significantly different to a random set of flock positions.

Method 2 vs. random - real world

The null hypothesis (H_0) is that in real world trials Method 2 shows results that do not differ significantly from those produced by a set of flocks randomly positioned in the arena.

The alternative hypothesis (H_1) is that the Method 2 (m2) controller performs better than random, ie. a sample from the population of m2 scores is likely to have a lower success score than a sample from the random population.

The data sets for this comparison are presented in Table 6.1 (bottom right). The samples of type $m1s$ are from the Method 2 real-world trials. The sample of type rnd are the distance-to-goal scores found for randomly generated points within the arena. The sample sizes are $n_1 = 7$ (m2r) and $n_2 = 7$ (rnd).

It is found that the U value for these data sets is 9. From the Mann-Whitney U test tables it is found that the probability of the two sets of samples coming from the same distribution is 0.027 (the U test tabels provide exact probabilities when the samples sizes are small, as in this case) . This is within the rejection range $\alpha = 0.05$. H_0 is therefore rejected in favour of H_1. It is shown that in the real world experiments Method 2 shows a significantly different average flock-to-goal distance to a random set of flock positions, and that the average is smaller for Method 2.

Method 2 vs. Method 1 - simulation

The null hypothesis (H_0) is that in simulation Method 1 shows results that do not differ significantly from those produced by Method 2.

The alternative hypothesis (H_1) is that the Method 2 (m2) controller performs better than Method 1 (m1), ie. a sample from the population of m2 scores is likely to have a lower success score than a sample from the population of m1 scores.

The data sets for this comparison are presented in Table 6.2 (left). The samples of type $m1s$ are from the Method 1 simulation trials. The sample of type $m2s$ are from the Method 2 simulation trials. The sample sizes are $n_1 = 12$ (m1s) and $n_2 = 12$ (m2s).

It is found that the U value for these data sets is 13. From the Mann-Whitney U test tables it is found that the critical U for a one-tailed test at $\alpha = 0.05$ is 42. H_0 is therefore rejected in favour of H_1. It is shown that in simulation Method 2 shows a significantly different average flock-to-goal distance to Method 1, and that the average is smaller for Method 2.

Method 2 vs. Method 1 - real world

The null hypothesis (H_0) is that in real world trials Method 1 shows results that do not differ significantly from those produced by Method 2.

The alternative hypothesis (H_1) is that the Method 2 (m2) controller performs better than Method 1 (m1), ie. a sample from the population of m2 scores is likely to have a lower success score than a sample from the population of m1 scores.

The data sets for this comparison are presented in Table 6.2 (right). The samples of type $m1s$ are from the Method 1 simulation trials. The sample of type $m2s$ are from the Method 2 simulation trials. The sample sizes are $n_1 = 9$ (m1s) and $n_2 = 7$ (m2s).

It is found that the U value for these data sets is 15. From the Mann-Whitney U test tables it is found that the critical U for a one-tailed test at $\alpha = 0.05$ is 15. H_0 is therefore rejected in favour of H_1. It is shown that in the real world Method 2 shows a significantly different average flock-to-goal distance to Method 1, and that the average is smaller for Method 2.

6.2 Purpose and limitations of the animal experiments

It should be noted that the experiments were designed to test whether the *robot* could achieve the goal task. They were not intended to assess any aspect of the *ducks'* behaviour other than tracking their position in the arena. For example no attempt was made to measure any testing order effects. In her experiments, Henderson found significant changes in the responses of the ducks to the same stimulus over three days of testing [Henderson, 1999, pp3.25-26]. It is reasonable to expect that there was some change in the behaviour of the ducks over time in these experiments. For example in the Method 2 further trials the same flock was used seven times over two days. It could be hypothesised that the ducks would become familiar with the experiment and that some change in their behaviour would be reflected in the success score recorded.

However the trials deliberately did not have the same start conditions; the position of the goal was randomised between trials. This was designed to *prevent* the flock from learning where the goal was located and thereby eliminate this source of order effect. A consequence of this is that the relative positions of the flock (which always entered through the same door) and the robot (which always started at the goal) at the start of the trial were always different, and are assumed to be random. These relative start positions can be assumed to have a strong effect on the outcome of the trial; for example if the flock starts very near the goal, a good success score can be expected. This gross effect is likely to obscure the effect of any subtle change in duck behaviour over time.

From the start, the behaviour of the ducks was *assumed* to change unpredictably over time and between flocks. This is the motivating assumption of the project. The experiments were designed to show that the robot system works *in spite of such variation.* Therefore the experiments deliberately put the robot and flock in a different relationship for each trial. It remains an interesting question as to how much variation is actually encountered in these scenarios, and this could be determined by repeating the experiments in identical start positions each time and observing any systematic changes in the system's behaviour from trial to trial.

6.3 Differences in simulated and real world results

The results in simulation are better than for the corresponding real-world experiments. There are four main reasons for this (in order of likely significance):

1. **Parameter optimization:** The parameters governing the behaviour of the flock control algorithms were chosen by experiment with the simulation. Once suitable parameters had been found, the simulation experiments were performed. The controllers were then transferred with identical parameters to the real world and tested on the real flocks. But the ducklets were (deliberately) not calibrated to best match the real ones, so the inevitable differences between the ducklets and the ducks meant that the controller was not optimally tuned for the flocks encountered.

   In particular it was found that the flock model parameter settings in the simulation caused the ducklets to move more slowly than the real ducks, so they traveled less distance in the 180 (simulated) seconds of the trials, reducing (improving) the efficiency score. The success score is likely to have increased (worsened) because the flock took longer to reach the goal position initially (described as the 'fetch phase' in Chapter 4).

   Therefore the real world trials were likely to achieve worse scores because the controllers were tuned for a different flock.

2. **Tracking errors (occasional):** The flock tracking software used in the experiments was unreliable (though it has since been improved (Appendix A)). The tracker would occasionally lose the flock and jump to an unrelated location. See for example Figure 4.21, where the flock is temporarily lost at around 22s. The path plot (A) shows the flock appearing to leap out of the top left of the arena. A corresponding jump can be seen on both the success and efficiency plots. These errors are incorporated into the average for the success plot, where they make a small contribution due to their short times pan. However they are simply added to the efficiency plot where they can make a more significant contribution to the final score. This will have caused the real world results to have been shifted to the right on Figures 6.3 and 6.4 compared to the simulation trials, by a (small) random amount in some trials.

3. **Tracking errors (systematic):** The vision system will also have added a more systematic bias to the efficiency score. The resolution of the sampled images (768x576 pixels - each pixel corresponds to approximately 1.3cm x 1.7cm of floor area)) and the small size of the ducks in the image produces a coarse sample of duck pixels. Even when the ducks are sitting still in the scene, variations in light and noise in the camera produce variations in the image, causing the detected flock centre to switch by a pixel or two. This 'wiggle' is recorded as a small movement by the tracker. The wiggle is also present when as the ducks move through the scene. Thus the ducks are measured to have moved further than they actually did, increasing (worsening) their efficiency scores; shifting all real world trials to the right on Figures 6.3 and 6.4. It may have been possible to determine an estimate of the contribution of the tracker wiggle and calibrate it out, but the system was dismantled before this effect was considered.

4. **Robot dynamics:** For simplicity the simulated system assumed zero time delays and perfect control response (infinite acceleration to a perfect speed) from the robot. Using the real system unavoidably introduced delays and control errors which will have slowed the response of the vehicle to the behaviour of the ducks. This may have reduced the effectiveness of the algorithms in the real world versus the simulation.

   The real robot dynamics could have been modeled with standard techniques after making accurate measurements of the time delays, robot mass, coefficient of friction, etc.. However, the discrepancies between the idealized robot model and the real robot were judged to be relatively insignificant compared to the other inevitable errors in the system, particularly the simulated/real flock discrepancies. A control algorithm which could accommodate these large differences would be unlikely to be very sensitive to smaller discrepancies in the robot dynamics. A much more accurate robot model would therefore achieve only a marginally more successful controller. These arguments are similar to those given for using a simple vision model, in section 3.7.2.

The results in simulation show less variation than in the real world. This is because the ducklets' response to the robot was identical in each trial. The variation in success/efficiency between trials was caused entirely by the different starting positions of the ducklets. The real ducks also varied in their start positions. In addition it is very likely that the real flocks showed some variation in their response to the robot, both from flock to flock and over time in the same flock (perhaps through habituation [Manning, 1972, p177], [McFarland, 1985, p316], etc). The variation between flocks and habituation are not examined here, though both are addressed at length in Jane Henderson's work [Henderson, 1999]. Also the real world experimental conditions could not be completely controlled, so the environment of the ducks (odour, sound, temperature, etc.) was slightly different in each trial with (possible) effects on behaviour.

6.3.1 Conclusions

It was shown that Method 2 achieves success scores that are significantly different to and better than random, and significantly different to and better than Method 1. This was the case both in simulation and in the real world. Method 1 was shown to be significantly different to and better than random in simulation, but in the real world this could not be shown.

It is suggested that Methods 1 and 2 would both achieve better performance scores if tuned by experiment with real ducks. However, this parameter tuning is undesirable in Animal Interactive Robotics applications, as discussed in Chapter 2. It is further suggested that Method 2 works better than Method 1 in the real world because it is less dependent on optimal parameter settings, but is intrinsically more reliable because of its 'proportional control' design. Thus it is the kind of method that is preferable for applications in Animal Interactive Robotics.

This section discussed the relative performance of two simple algorithms which solved the flock gathering task with varying degrees of success and efficiency. Two further experiments were performed to examine the behaviour of the flock in the arena in the absence of the robot. Results from these control trials are used as benchmarks to further examine the efficacy of the algorithms.

6.4 Control 1: no stimulus

In these first control trials the flock is tracked as it moves freely in the arena with no robot present. The aim is to establish that the flock does not move to the goal position without the stimulus presented by the robot.

6.4.1 Experimental procedure

The experimental flock is initially held for three minutes in a small pen with a door that opens into the arena. At the beginning of the trial the door is opened and the flock is tracked as it moves freely around the arena for the next three minutes. The arena is empty but is not specially prepared or cleaned between trials. The point on the arena wall opposite the entrance door is nominated the goal position. As in all the trials so far, there is no physical indication to the ducks of the goal position.

The trial is performed once for each of three flocks. The flocks used are the same three flocks of 12 ducks used in the Method 1 trials. This experiment differs from the previous ones in that the ducks are not allowed a three minute settling time after introduction into the arena. For direct comparison with the other trials it may have been more appropriate to have included this settling time. However, as the flocks typically settle in a small part of the arena and don't move about much after this period, it was decided that the first three minutes movement would be more interesting to record.

The usual path, success and efficiency data are recorded and presented below.

6.4.2 Results

Figures 6.5 to 6.7 show the results of the no-stimulus control trials.

Figure 6.5 (A) shows the path of the first flock as it comes out of the door at the top of the plot and moves to the left of the arena, where it spends most of the trial. The spikes on the distance-to-goal plot (B) at around 100s correspond to the glitches on the path plot that appear to show the flock jumping out of the arena at (-3,3); this is caused by temporary failures in the flock tracker. These errors generate false distance-traveled data; the sharp increase at 100s in the plot (C) is caused by this error. The distance-to-goal plot shows that the flock does not approach the goal at (0,-3.5).

The path plot for flock 2 (Figure 6.6 (A)) begins at approximately (0.5,1.2) rather than the top of the arena. This is due to a procedural error whereby the flock was let in to the arena approximately 3s before the position data began to be logged. The flock in fact moved in a roughly straight line from the entrance door at (0,3.5) to the first logged position in this time.

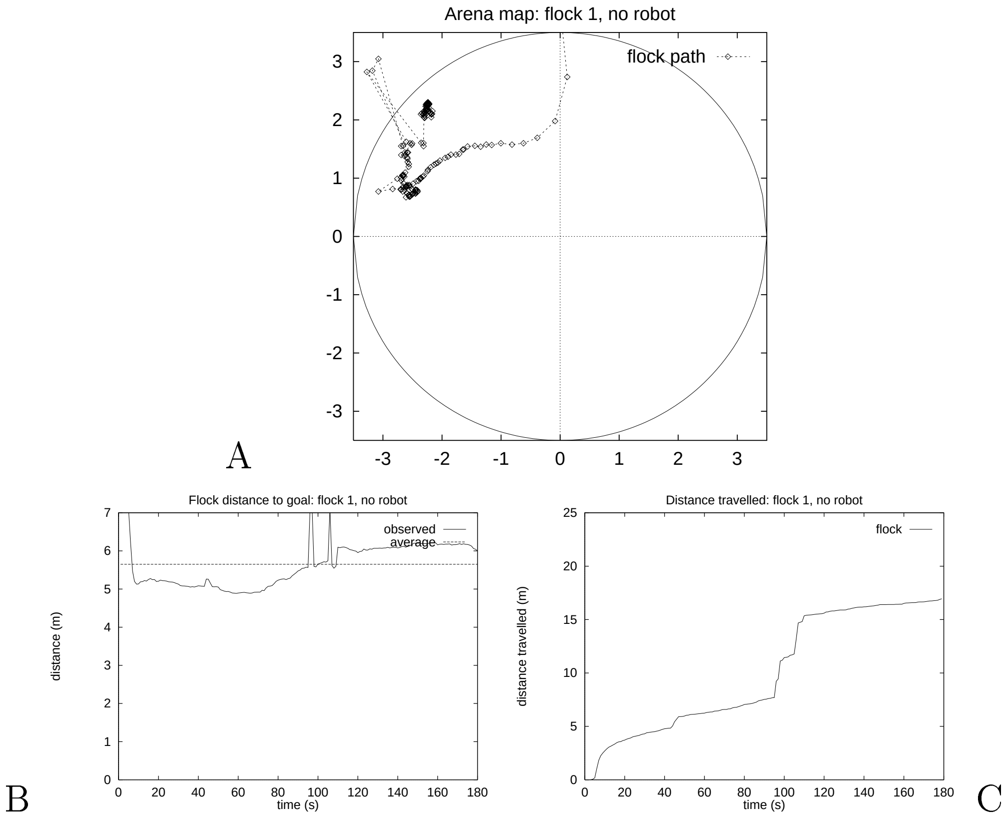


Figure 6.5: Control 1 results - no stimulus - flock 1.

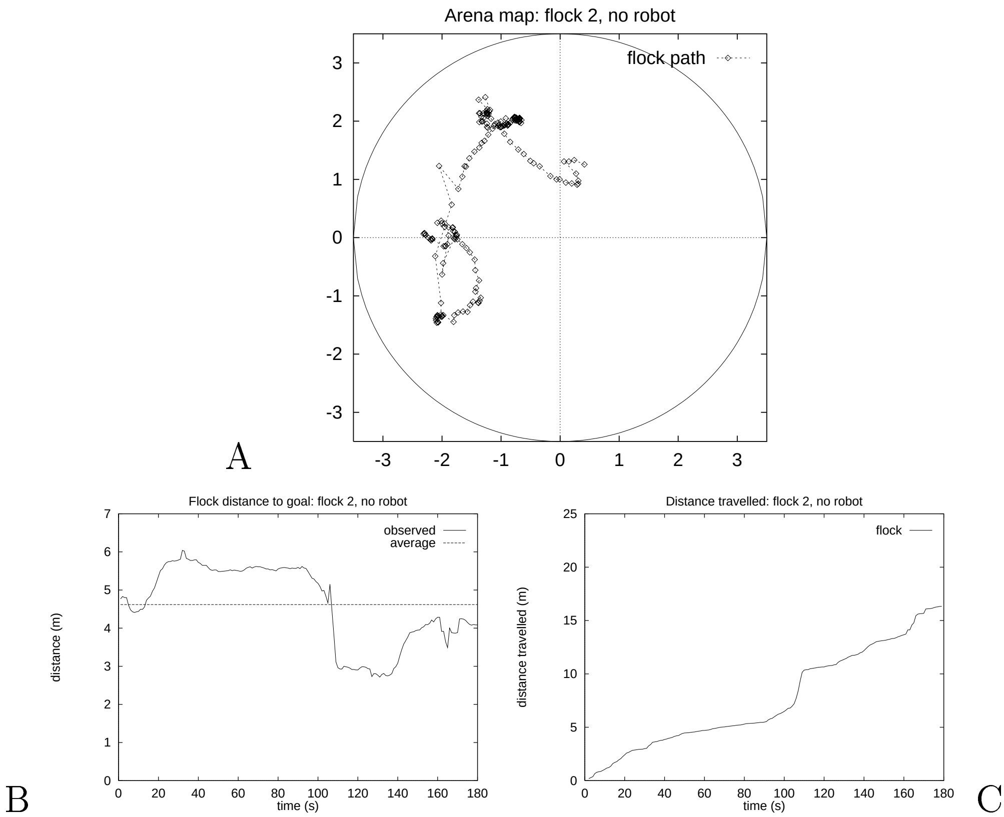


Figure 6.6: Control 1 results - no stimulus - flock 2.

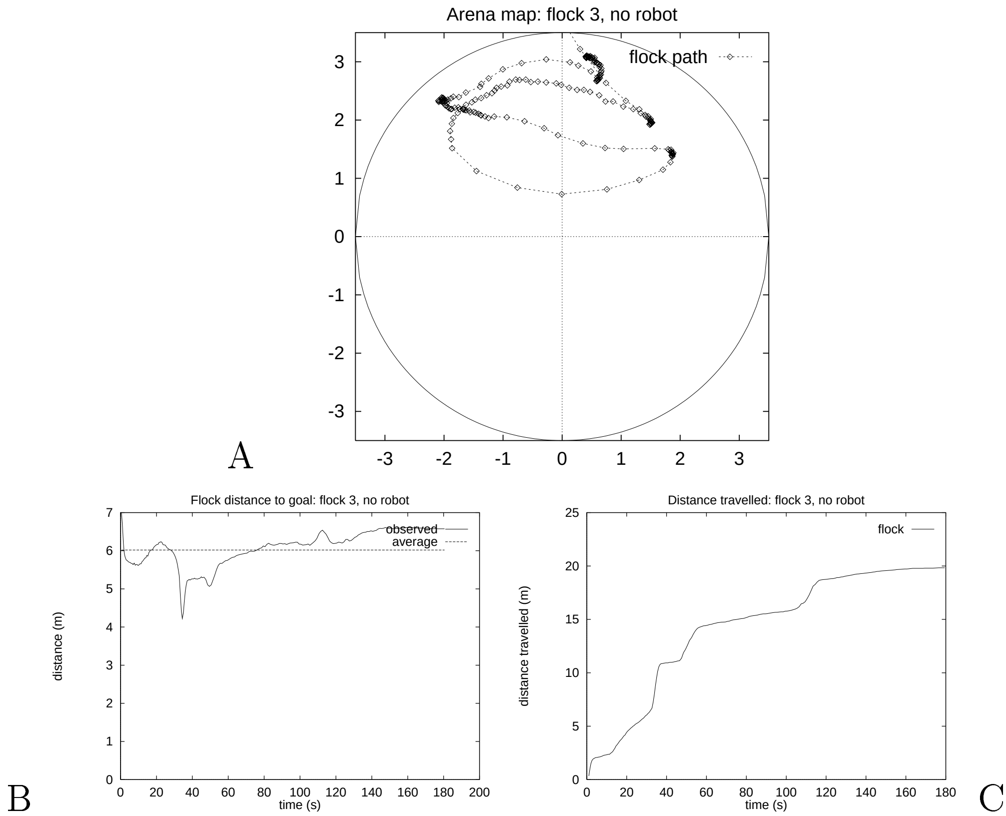


Figure 6.7: Control 1 results - no stimulus - flock 3.

Trial	Efficiency	Success
1	16.95	5.65
2	16.33	4.62
3	19.83	6.02
total	53.11	16.29
average	17.70	5.43
variance	3.49	0.53
stdev	1.87	0.73

Table 6.3: Summary of results for no-stimulus control trials All results are measured in metres.

Table 6.3 shows the final scores from these trials. It is recognized that the very small sample size means that the statistics are not convincing if considered on their own. However, when read along with the qualitative results presented in the plots, the reader should be satisfied that the flocks do not tend to move towards the goal position without the stimulus provided by the robot. A poor success score is thus recorded, as could be expected.

These results also present a baseline for the amount of distance traveled by the flock when introduced into the arena but with no other stimulus. Recall that distance traveled was assumed to be (crudely) indicative of duck stress when defining the efficiency measure in Chapter 4. A good efficiency score is recorded, as the flock has little motivation to move. The overall distances traveled by the flocks in all the experiments are presented below.

6.5 Control 2: food stimulus

The previous control experiment was designed to set a null baseline to compare with the robot trials. To complement this a further experiment was designed to set a positive baseline. The idea was to attract the ducks to the goal by providing some positive stimulus; the 'carrot' as opposed to the 'stick' of the robot trials.

6.5.1 Experimental procedure

The experimental procedure was identical to that for the first control above, except that a bowl of the ducks' normal food was placed as close as possible to the goal position. The bowl was placed just far enough away from the arena wall to allow room for the ducks to eat from all sides. The trials were performed first thing in the morning, when the ducks had not been fed since the previous evening. Jane Henderson had found previously that the ducks were well motivated to eat in these conditions.

The same three flocks of ducks were used as for the previous control experiment, and the same data were recorded.

6.5.2 Results

Figures 6.8 to 6.10 show the results from the food-bowl control trials.

Flock 1 enters the arena through the door at the top of the plot ((0,3.5) on Figure 6.8 (A))

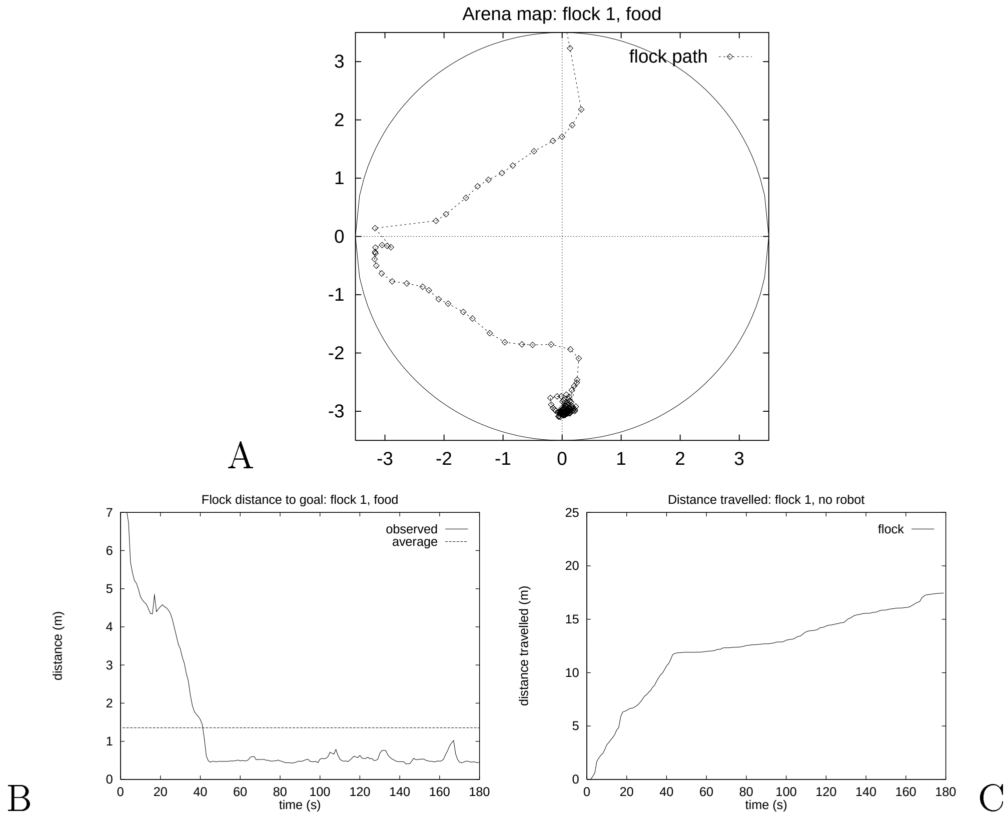


Figure 6.8: Control 2 results - food bowl - flock 1.

and moves down and to the left of the arena, pausing briefly at around 20s (success plot (B)). They then move quickly to the food bowl by the goal at around 40s and stay close to it for the rest of the trial. The efficiency plot (C) shows that the flock speed changes suddenly when the flock reaches the food at 40s. The subsequent movement recorded on the plot is generated by the flock milling around the food bowl, competing for access to the food.

Flock 2 heads more directly to the food (Figure 6.8 (A)) and shows correspondingly good success and efficiency plots. Flock 3 shows initial behaviour some way between flocks 1 & 2, but the flock does not spend its time so close to the goal, and wanders off to the middle of the arena near the end of the trial (170s).

Table 6.4 shows the final scores for these trials. Again the statistics suffer from the small sample size, but should be considered along with the qualitative records of the behaviour shown in the plots. The food bowl control scores very well for success, and with an efficiency similar to that recorded for the no-stimulus control.

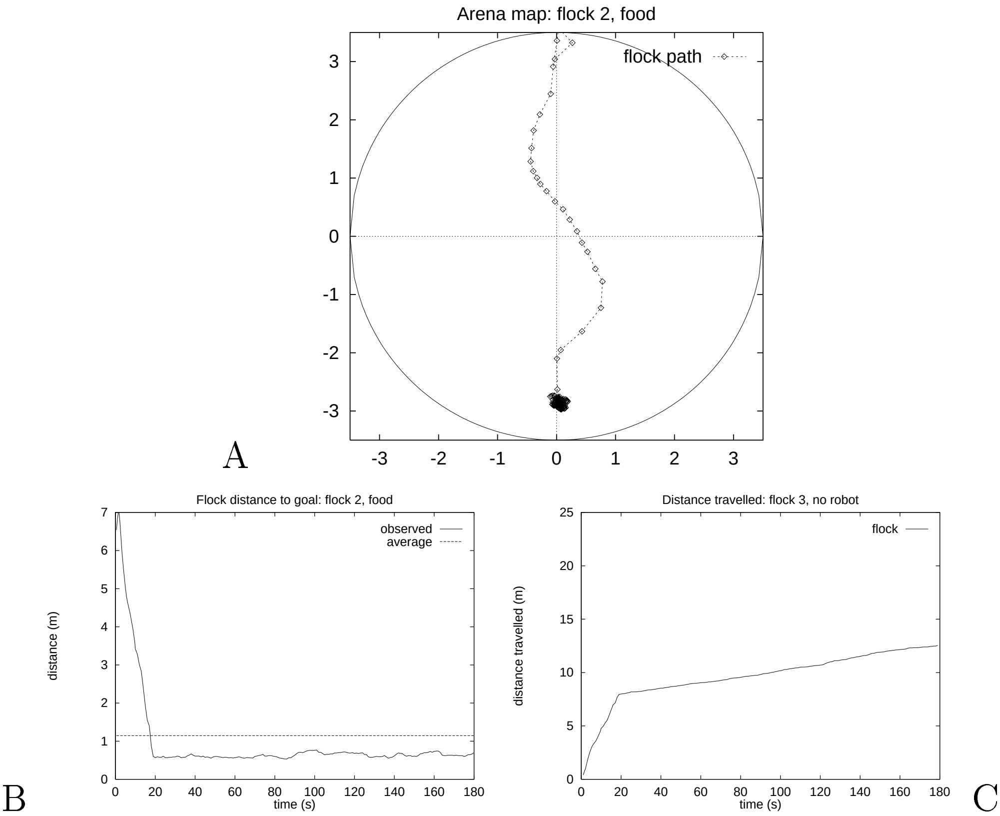


Figure 6.9: Control 2 results - food bowl - flock 2.

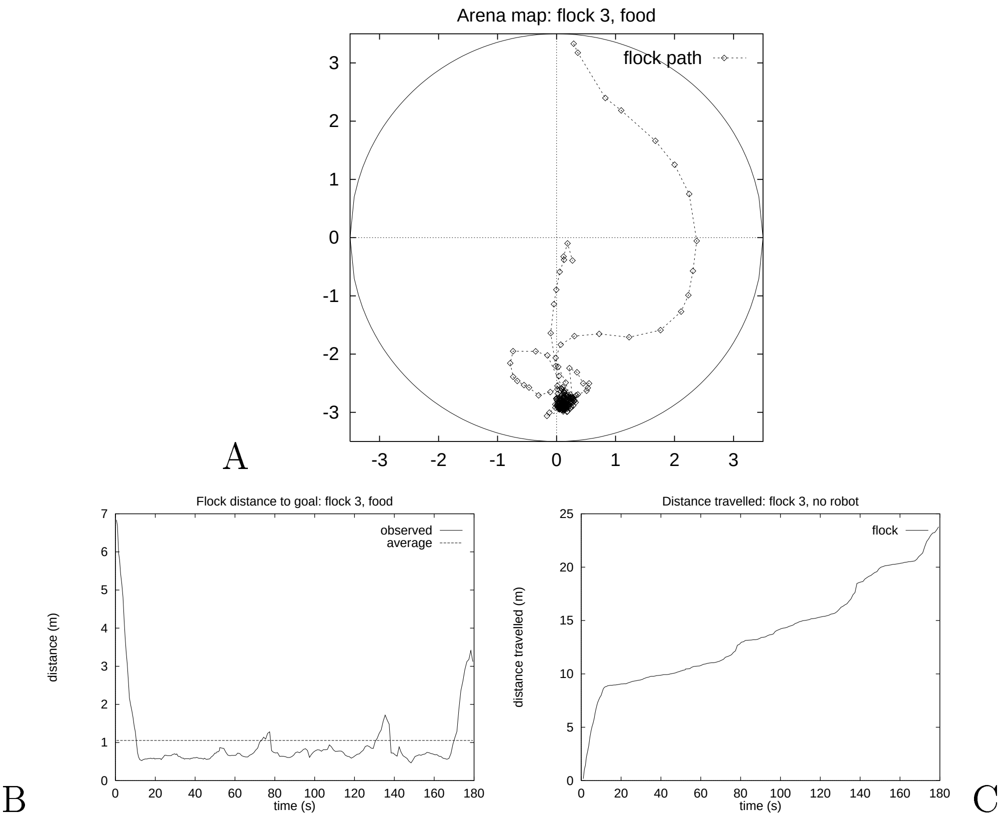


Figure 6.10: Control 2 results - food bowl - flock 3.

Trial	Efficiency	Success
1	17.46	1.36
2	12.55	1.14
3	23.78	1.05
total	53.79	3.55
average	17.93	1.18
variance	31.69	0.03
stdev	5.63	0.16

Table 6.4: Summary of results for food bowl control trials. All results are measured in metres.

6.6 Comparing the control and real results

Results from the Method 1 & 2 real world trials are re-plotted with the controls in Figure 6.11. This time the success score is plotted against the distance moved by the flock alone, not the efficiency (flock+robot distance) used previously. The flock movement distance can be compared with the control trials in which there is no robot component to add. By ignoring the movements made by the robot the overall success of the system at gathering ducks can be seen and compared to the no-stimulus and positive-stimulus baselines provided by the controls. The boxes indicate the range of the results observed.

Figure 6.12 shows the averages of the points plotted in Figure 6.11. Again the length of the vectors from the origin to the average points can be used as an indication of 'goodness'. The boxes indicate ± 1 standard deviation from the mean (though the same caveats apply as for Figure 6.2).

The Control 1 no-stimulus trials achieve the poorest success score, but a good flock movement score. The ducks were not very motivated to move, and they didn't move to the goal.

Method 2 is again shown to be superior to Method 1 both in terms of flock distance-to-goal and flock distance-traveled. Method 2 also shows less variation between trials, suggesting better reliability.

The ducks spent the most time close to the goal and traveled the least distance in the Control 2 food-bowl trials. Attracting the ducks with food outperformed the robot control methods, though Method 2's performance is not far off. It could be concluded that attracting the flock with food is the best approach for solving this task. However using the robot might allow more scope for performing more complex tasks, such as moving the flock along a specified path over time.

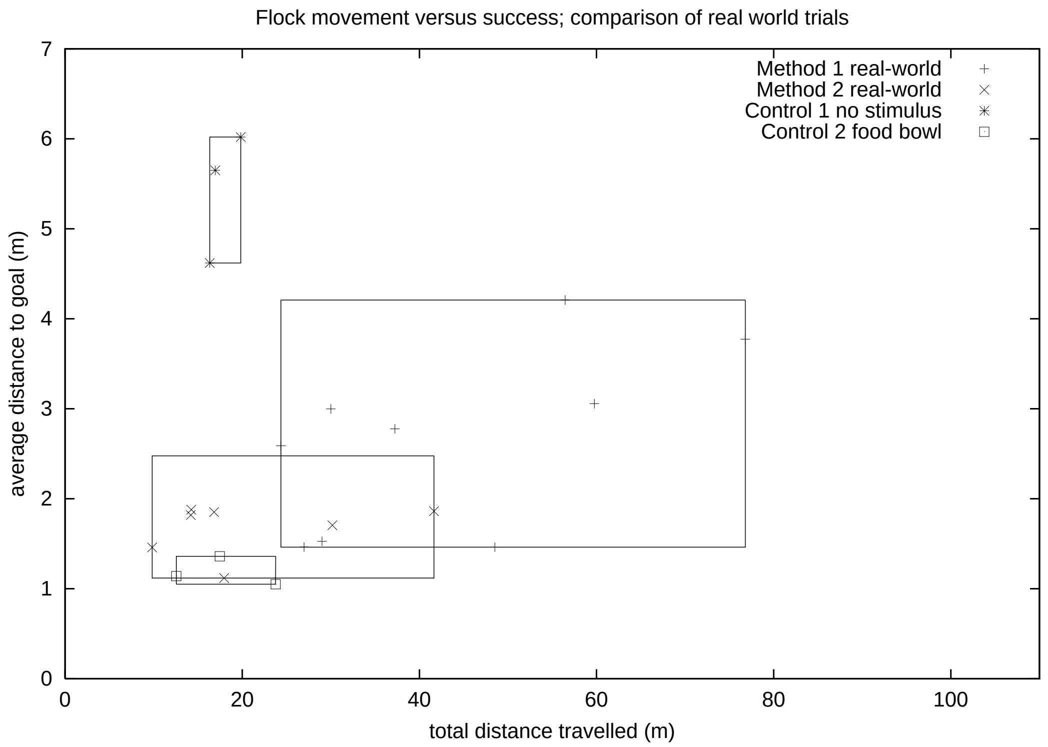


Figure 6.11: Distribution of success versus flock movement scores in the control and real world trials. Boxes indicate range of results.

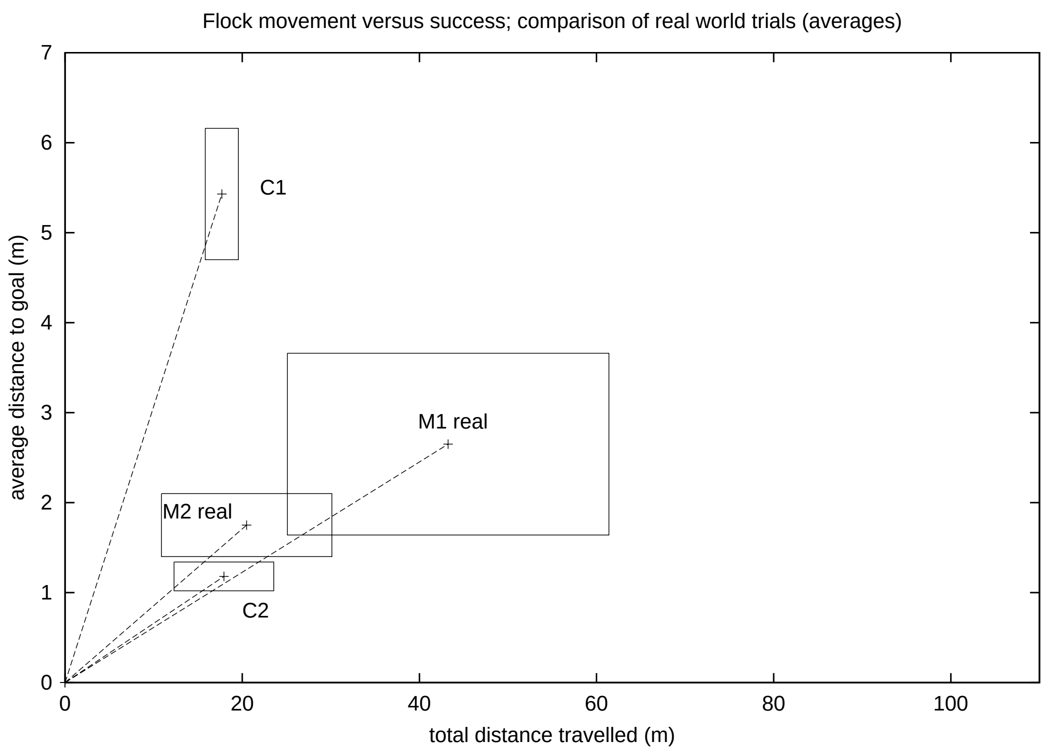


Figure 6.12: Mean success/flock-movement scores. Boxes indicate standard deviation from the mean.

6.7 Questions, limitations and extensions

There are are many possibly interesting variations on the experiments presented in this thesis. A common form of question at conference presentations has been "What happens if there are corners in the arena / the goal is away from the wall / you use more robots / etc.". This section briefly covers some of these issues, though undertaking further experiments to explore this particular application in detail were not considered essential to this thesis, and in any case were precluded through lack of time. Some possibilities for further work are also discussed in Chapter 7.

6.7.1 How would you tackle corners?

The experimental arena was designed with no corners to keep the task as simple as possible. A more realistic arena, perhaps modeled on a poultry house or farmyard, would probably include corners.

Corners are a problem for potential-field based controllers; they are a common source of local minima. This is essentially the same problem faced by the 'cornered animal': faced with a threat in one direction, but surrounded by walls, there is no 'good' direction to move in. As the threat approached, the stimulus (potential) increases but the best place to be (local minimum) remains in the corner. This continues until finally the animal must move *towards* the threat for a short time, (which may even *increase* its stimulus), in order to attack (which we will discount), or to get to a lower-threat region away from the corner. As it runs past the threat, the stimulus is typically very strong, so the animal will be moving quickly and may be panicked, etc. This is undesirable in an Animal-Interactive Robotic system as it implies reduced animal welfare and loss of effective control.

Tackling a 'cornered animal' is then a challenging task for a robot system, but two few basic observations can be made that may help the design of a corner-capable system.

- To ensure an animal or flock moves out of a corner, the robot must move in to the corner. Assuming the robot doesn't totally physically block their escape, they will eventually move.
- If the robot approaches from one side rather than the middle of the corner, then the animal/flock is likely to leave by the other, open side.

The Method 2 controller exerts a potential that attracts the robot to the flock. It does not have Method 1's robot-flock repulsion. Also the robot's goal-repulsion potential biases the robot to be positioned behind the flock with respect to the goal. Thus the Method 2 controller would cause a robot to enter the corner towards the flock on the side opposite the goal; pushing the flock out of the correct side nearest the goal. This is the desired behaviour, but it is easily defeated by a the local minima problem in an arena where the flock must be moved away from the goal for some distance in order to finally reach it.

This problem could be tackled by planning a route for the flock to take to the goal, producing a series of 'way-points'; sub-goals that the flock should reach on its way to the goal. A way-point would exert a potential on the robot until reached by the flock, just as the main goal does in these experiments. A series if way-points could be activated in turn until the flock reaches the main goal. Planning such paths quickly and reliably, and particularly guaranteeing them to be free of local minima is a research topic all of its own and is not discussed further (though see [Hague et al., 1990] for a contribution by a member of this project).

6.7.2 What happens if the goal is away from the wall?

It can be seen from all the (successful) trials that the flock is effectively held near the goal *against the arena wall.* How would the system behave if the goal is not near the wall?

The two Methods presented rely on the presence of the wall to restrict the movement of the flock about the goal. This approximates the canonical one dimensional control problem of balancing a stick on a cart. The need for this restriction arises from the fact that the robot can only apply a 'pressure' on the flock in one direction at a time.

For both Methods as the distance from the goal to the wall increases, the ability of the robot to balance the flock at the goal decreases. The likelyhood of the system stabilizing with the flock near the goal reduces, and a new behaviour emerges whereby the flock 'orbits' the goal position, chased by the robot. The orbiting can be thought of as the two-dimensional equivalent to the oscillation about the goal observed with Method 1 as the flock moves along the one-dimensional surface of the wall.

If there is no wall at all, ie. the robot is acting on an unconstrained plain, then Method 1 will fail to contain the flock in a finite area due to the robot/flock repulsion. In the constrained arena this potential prevents the robot getting too close to the flock. On the unconstrained plane, as soon as the robot lies between the goal and the flock, it can never get to the other side of the flock to push it back to the goal. This is because the robot/flock repulsion will dominate the controller as the robot approaches the flock; the robot can never quite reach the flock, let alone get past it. The flock is therefore chased away into infinity. Of course, this outcome only applies to the model agents in the abstract; the real robot and flock are always physically constrained in some manner.

Method 2 would be more successful because no robot/flock repulsion is applied. If the robot can move faster than the flock then the controller will keep the flock a finite distance from the goal - the flock will never 'escape' the robot and run off into infinity. The robot will repeatedly chase the flock over the goal position, catch up and pass through it, and chase it back again. The size of the area in which the flock is held is a function of the relative speeds of the flock and robot. These depend both on their respective 'controllers' and on mechanical speed constraints.

It is clear that oscillation of the flock about the goal position is a bad result according to the welfare requirements described previously. The ideal result is for the flock to settle at the goal. Detailed examination of the outcomes described above was beyond the scope of this project, but it is worth noting that in the real-world Method 1 and 2 trials, oscillation was less apparent than in the simulation. The Method 2 experiments showed that reducing the robot's distance from ducks reduced their motivation to move very effectively, up to the point where the ducks would sit down and not move at all, which of course precludes any oscillation. This behaviour is not modeled in the simulator but should be straightforward to include. Carefully timing the robot's retreat from the flock might allow control of the position at which the ducks stopped moving and sat down. This could provide a basis for further work to develop successful methods for holding a flock in space with a single robot.

[No examples were available for this section, as the simulator was no longer available as this was written. The discussed behaviour had been observed previously].

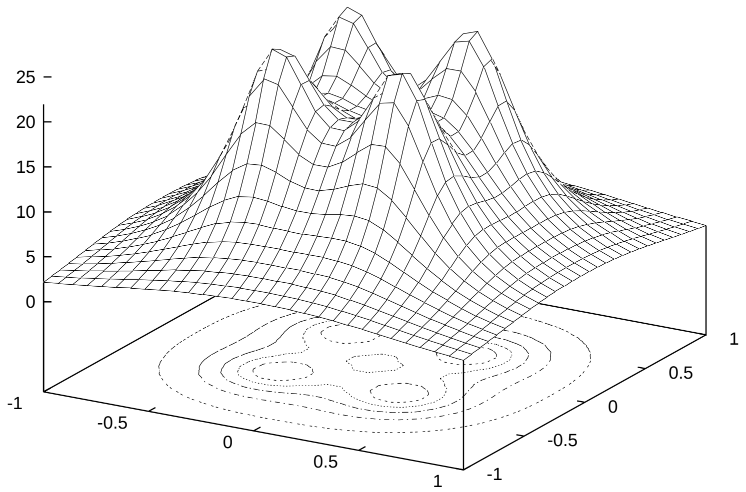


Figure 6.13: Map of repulsive potential when four robots are equally spaced around the flock at (0,0). A local minimum exists at (0,0) between the peaks, which could be used to contain the flock.

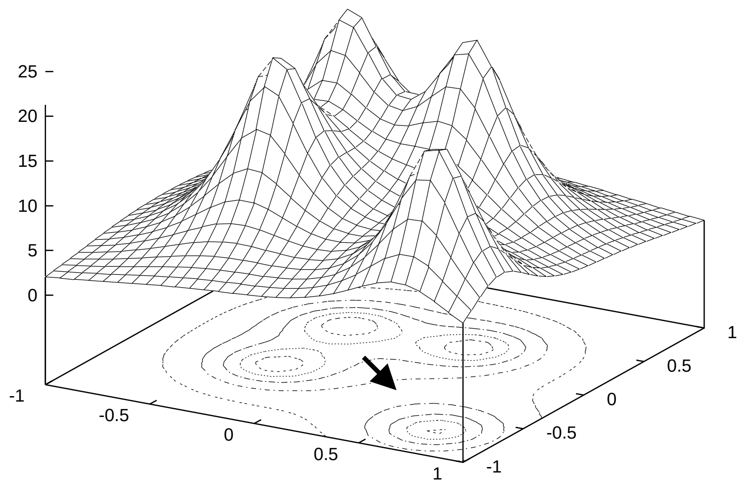


Figure 6.14: Map of repulsive potential when four robots surround the flock at (0,0), but one robot is further from the flock. A potential gradient exists at (0,0) which would drive the flock in the direction of the arrow.

6.7.3 Could you use more than one robot?

Multiple robots could offer a way to constrain the flock in free space. To 'balance' the flock at a goal in free space at least *three* sources of pressure on the flock are required. Three or more objects exerting a potential on the flock can enclose a polygon containing a *potential minimum* which would constrain the flock.

An example of a four-robot potential field is given in Figure 6.13. The map surface shows the magnitude of potential that would experienced by a ducklet at each point. It is assumed that each robot makes a similar contribution to the flock's movement $\vec{F}$ according to some function of the flock-to-robot vector $\vec{FR}$. The final flock movement vector $\vec{f}$ is given by the sum of all the robot potentials:

$$\vec{f} = FR_1 + FR_2 + FR_3 + FR_4$$

The figures are plot of the inverse-square-of-distance function

$$z = \frac{1}{({x_1}^2 + {y_1}^2) + s} + \frac{1}{({x_2}^2 + {y_2}^2) + s} + \frac{1}{({x_3}^2 + {y_3}^2) + s} + \frac{1}{({x_4}^2 + {y_4}^2) + s}$$

Where s is a small positive value to keep the peaks finite for visualization purposes.

If a ducklet (or a flock of ducklets) was inside the square formed by the robots, then they would move to the local minimum of potential at (0,0). The ducklet would be held there indefinitely, as ducklets will never move up a potential gradient.

If the four robots were to move in synchrony in any direction, then the ducklet(s) would be compelled to move in the same direction in order to minimize the potential on them. This could offer a way of moving an agent or flock over a prescribed *path* instead of just holding at a *point* as discussed in this project.

Another possibility with this configuration is to move one robot out of the formation, further from the ducklets. The resultant on the ducklets is then towards the most distant robot. This is shown in Figure 6.14 with the direction of the resultant on the ducklet shown by the arrow. This more complex way of controlling the direction of movement of the ducklets could form the basis of a more sophisticated flock-controller.

6.7.4 Overhead camera worries

The overhead camera gives a view that is not available to a sheepdog. It would be more interesting to have the sensing on the robot.

This was an interesting point, and while the author is happy to describe the Robot Sheepdog system as 'autonomous' in the sense of 'self-guided', some commentators prefer that an autonomous robot should be self-contained, with diectic (local, self-referenced [Agre and Chapman, 1990]) sensing rather than any global scheme as used here.

This was raised so often that it seemed worthwhile addressing. It also presented a challenge, for while the control algorithms (Methods 1 & 2) were expressed only in the *relative* positions of the robot, flock and goal and thus partly satisfied the diectic requirement, they indeed relied on the input from an *absolute* sensing system. It was not immediately clear that they would continue to work when some information was lost by adopting a local dog's-eye-view sensing scheme. If they did transfer to local sensing it would add support for their generality and effectiveness.

The robot's sensing requirements are fairly modest. Both controllers require a continuous estimate of the distance and direction to the flock center. In addition, Method 2 requires a measurement of the robot's distance to the goal from which it can calculate the flock-to-goal distance by triangulation.

6.8 On-board sensing

This section demonstrates that the developed flock-control techniques are applicable to on-board local sensing modalities.

Two local sensing examples are presented, the first using a rangefinder sensor, the second with an onboard camera. Both are examined only in simulation, as real world experiments were beyond the scope of this project due to time and cost. The previous experiments have shown the successful transfer of similar controllers from simulation to reality. While transferring the techniques used in these examples to the real world would not be as straightforward as the original experiments, they have been designed to a make such as transfer as simple as possible. The sensor requirements are feasible, as described in the introductions to each method.

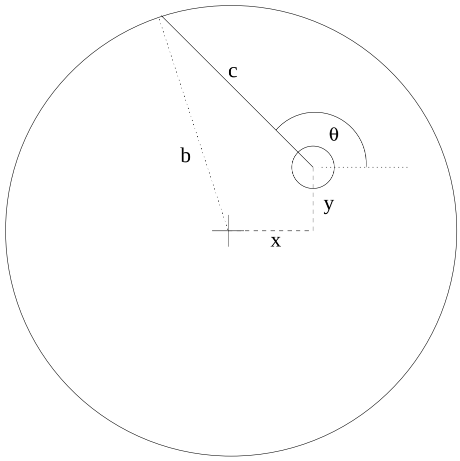


Figure 6.15: Finding the range c along a ray at angle θ to the arena boundary for a robot at (x, y) in an arena of radius b.

6.8.1 Rangefinder method

This example simulates an onboard rangefinder device, providing range to nearest object over the full 360^o field. Real scanning laser rangers such as the commonly used SICK laser (http://www.sickoptic.com/) typically have a field of view of close to 180^o, so this arrangement could achieved using two devices. Sonar rangers are often used in robot experiments due to their relative low cost and power requirements. However sonar suffers from poor resolution, noise and interference problems, while this example relies on relatively high-resolution range measurements. It therefore corresponds more closely to the abilities of laser rangers; the SICK laser rangers are claimed to provide a resolution of up to 0.25^o with an accuracy of 0.01cm (http://www.sickoptic.com/laser1.htm).

Generating the sensor data

The robot is provided with a vector of 256 range measurements, corresponding to the distance to the nearest surface (ducklet or wall) along each of 256 ‘rays’ traced in evenly spaced directions from the robot’s centre.

Figure 6.15 shows the variables required for the range calculation. Given the position of the robot relative to the arena centre (x, y) and the arena radius b, the range c from the robot centre to the arena wall along a ray at angle θ can be calculated. Again, all angles are measured in radians.

It can be seen from Figure 6.16 (i) that side c forms a triangle with the sides a and b. b is the

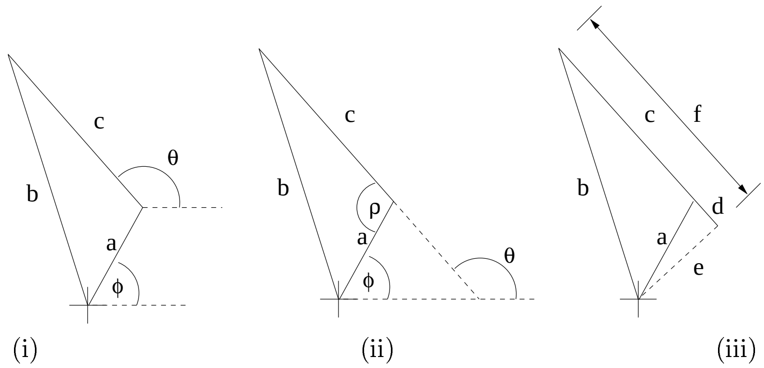


Figure 6.16: Illustration of intermediate steps in calculating the length of ray c.

radius of the circle and is known, while a can be found by:

$$a = \sqrt{x^2 + y^2}$$

The angle ϕ between side a and the x-axis is given by:

$$\phi = \arctan\left(\frac{y}{x}\right)$$

The angle ρ opposite b can be determined from θ and ϕ, illustrated in Figure 6.16 (ii):

$$\rho = \pi + \phi - \theta$$

Now the projection of side a onto side c is found, to obtain distance d:

$$d = -a \cos \rho$$

Side d forms a right-angled triangle with side a and side e, as shown in Figure 6.16 (iii). The length of c can now be determined thus:

$$\begin{aligned} e^2 &= a^2 - d^2 \\ f &= \sqrt{b^2 - e^2} \\ c &= f - d \end{aligned}$$

The range c is quantized and normalized so it can be stored as a one-byte value (ie. integers from 0 to 255) where 0 is 0m and 255 is the arena diameter 7m - the largest range value possible. This gives a resolution of 7/256m = 0.027m.

The range for each ray is calculated in turn and the value filled into the vector until the vector is entirely filled with range-to-wall values.

Next the range readings to the ducklets are calculated and superimposed over the filled array. For each ducklet, the distance d from the robot at (x, y) to the ducklet at (x', y') is calculated.

$$d = \sqrt{(x - x') + (y - y')}$$

The angle ϵ from the robot to the centre of the ducklet is given by:

$$\epsilon = \arctan\left(\frac{y - y'}{x - x'}\right)$$

The angle ϕ subtended on the robot's visual field is calculated from the range d and the ducklet's radius r:

$$\phi = 2\arctan\left(\frac{r}{d}\right)$$

In practice these angle calculations must take account of the signs of their arguments to give the resulting angle in the correct circle quadrant. The simulator code uses the standard C library's 'atan2()' function to take care of this. When repeatedly adding angles, the code takes the modulus of the sum and 2π radians to effectively 'wrap-around' the result.

Finally the elements of the range vector corresponding to the angles $\epsilon - \frac{\phi}{2}$ to $\epsilon + \frac{\phi}{2}$ are examined. If the value of any of these elements is greater than d, ie. it has a range reading corresponding to an object further away than this ducklet, then that element is set to d. Once this process has been performed for all ducklets, the range vector is complete. These simulated range readings do not account for the *shape* of the ducklets, but treat them as a flat surface perpendicular to the ray angle. While this would not accurately model real range readings, this simplification is not relied upon for the controller to work properly, and should not effect the ability of the controller to work with real ranges.

Controller

Given the complete range data vector described above, along with the direction of the goal position, the robot is guided by a a modified version of Method 1. This work was performed before Method 2 was devised, but Method 2 could easily be adapted to work in this case and could be expected to produce better results.

First the range data is analyzed to identify and label samples as corresponding to walls and non-walls. As there are no other objects in the arena, any sample which is judged to not be a wall

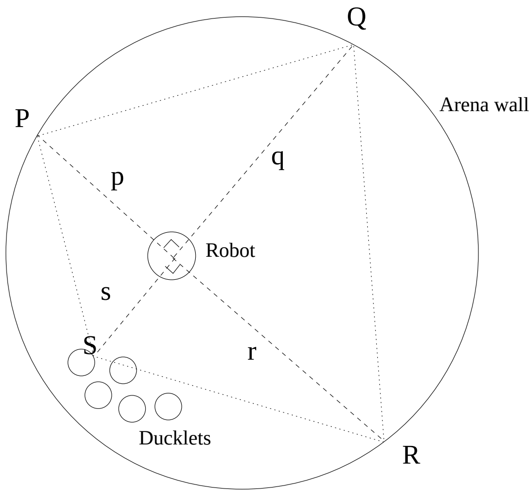


Figure 6.17: Categorizing range reading as referring to wall or ducklets using 4 perpendicular range readings. One triangle (in this case PQR) has vertices which lie on a circle with the same diameter as the arena. The range reading that corresponds to the ducklets must be opposite Q.

is assumed to relate to a ducklet. The following method was devised:

Each set of 4 perpendicular ranges (p, q, r, s) is considered in turn, as in Figure 6.17. For each of the triangles PQR, QRS, RSP, SPQ, the diameter of the circle on which the vertices fall is given by the ratio of the product of the two shorter sides to the height of the triangle:

$$f(p,q,r) = \frac{\sqrt{q^2+r^2}.\sqrt{q^2+p^2}}{q}$$

This formula is derived from the general method for finding the *circumcenter* of the triangle; the center of the circumscribing triangle which is also the point of equal distance from all three corners. That distance is the radius of the circle [Weisstein, 1998].

If all three points of a triangle fall on the arena wall then the diameter recovered will be $\approx$ 7m, with an error in line with the resolution of the original rangefinder sensor. The system compares the diameter obtained to the known diameter of the arena, and if they match to some threshold then all three ranges can be labelled as WALL ranges.

If the circle diameter differs from the known arena diameter by greater than some threshold amount, then at least one of the three ranges *must* correspond to a ducklet. To determine which one we make a simplifying assumption that it is *only one*. This is justified by considering the behaviour observed in the real-world experiments where the flock never splits up and the robot does not get very close to the flock. Therefore range samples separated by $\geq 90^o$ are unlikely to both contain ducklet-related samples.

Given that there can be only one ducklet range per set of four samples, there must be one triangle whose vertices do lie on a 7m circle, eg. PQR in Figure 6.17. The ducklet range is therefore the sample *opposite* the middle vertex (Q) of that triangle. The opposite sample can be labelled as a DUCKLET range.

Now that the ranges are labelled, a movement vector $\vec{r}$ can be generated: The robot is (1) **attracted** in the direction of DUCKLET labelled samples, with a magnitude proportional to the range. It is also (2) **repelled** from the direction of WALL labelled samples by an amount inversely proportional to distance, preventing the robot from hitting the walls. A final potential (3) **repels** the robot from the goal at point G by a constant amount. Considering the range samples as vectors $\vec{d}_{d0\to s_1}$ labelled DUCKLET and $\vec{d}_{w0\to s_2}$ labelled WALL , this can be expressed as:

$$\vec{r} = \underbrace{\sum_{n=0}^{n=s_1} K_1 \vec{d}_{dn}}_{(1)} - \underbrace{\sum_{m=0}^{m=s_2} \frac{K_2}{|\vec{d}_{wm}|}\, \widehat{d}_{wm}}_{(2)} - \underbrace{K_3 \widehat{RG}}_{(3)}$$

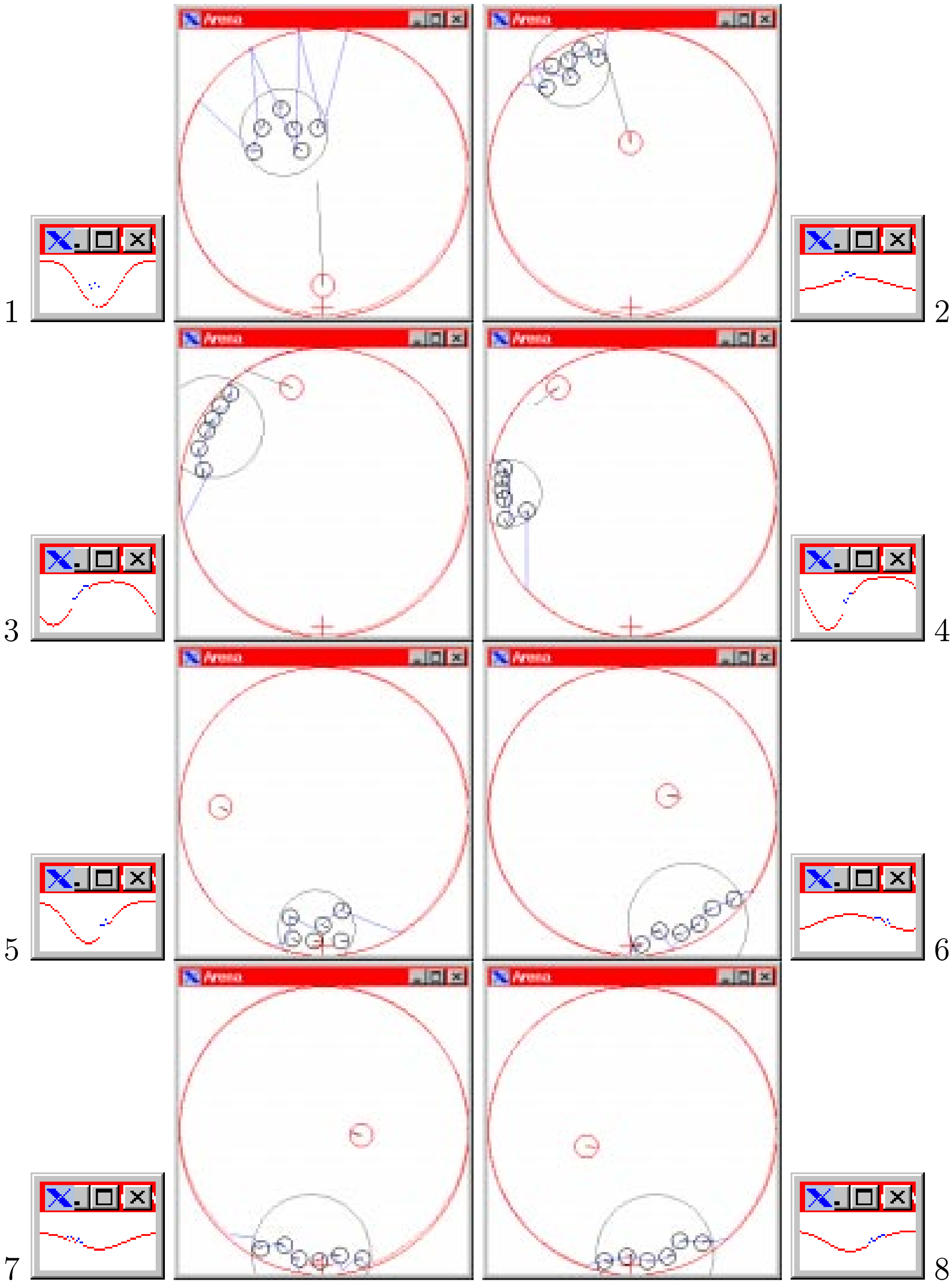
It is assumed that the robot-to-goal unit vector $\widehat{RG}$ is obtained from a separate system which is not discussed here. No such assumption is made in the next example.

The gain parameters $K_{1\to 3}$ were adjusted by experiment until a satisfactory behaviour was observed.

Results

Figures 6.18 and 6.19 shows the results from a trial identical to those performed for Methods 1 & 2 in simulation. Figure 6.18 shows screenshots from the simulator as the trial runs, numbered in order. The large boxes show the state of the arena with the synthesized range samples superimposed on the arena. The small boxes show graphs of the range sample vectors, with the discontinuities in the curve of the graph corresponding to ducklet-related samples.

The path plot (Figure 6.19 (A) shows that the flock starts out in the top left quadrant of the arena, with the robot near the goal at the bottom. As the trial runs, the robot moves towards the flock which is pushed towards the wall. The robot moves behind the flock with respect to the goal and pushes it down the wall to wards the goal. As the flock approaches the goal, the robot



Key: ◯ = robot, ◦ = ducklets, + = flock goal. Range readings shown by continuous dotted line. Small windows show graph of range readings from $-\pi$ to $+\pi$ radians relative to robot heading.

Figure 6.18: Sequence of images from the simulator during a rangefinder trial, showing successful behaviour.

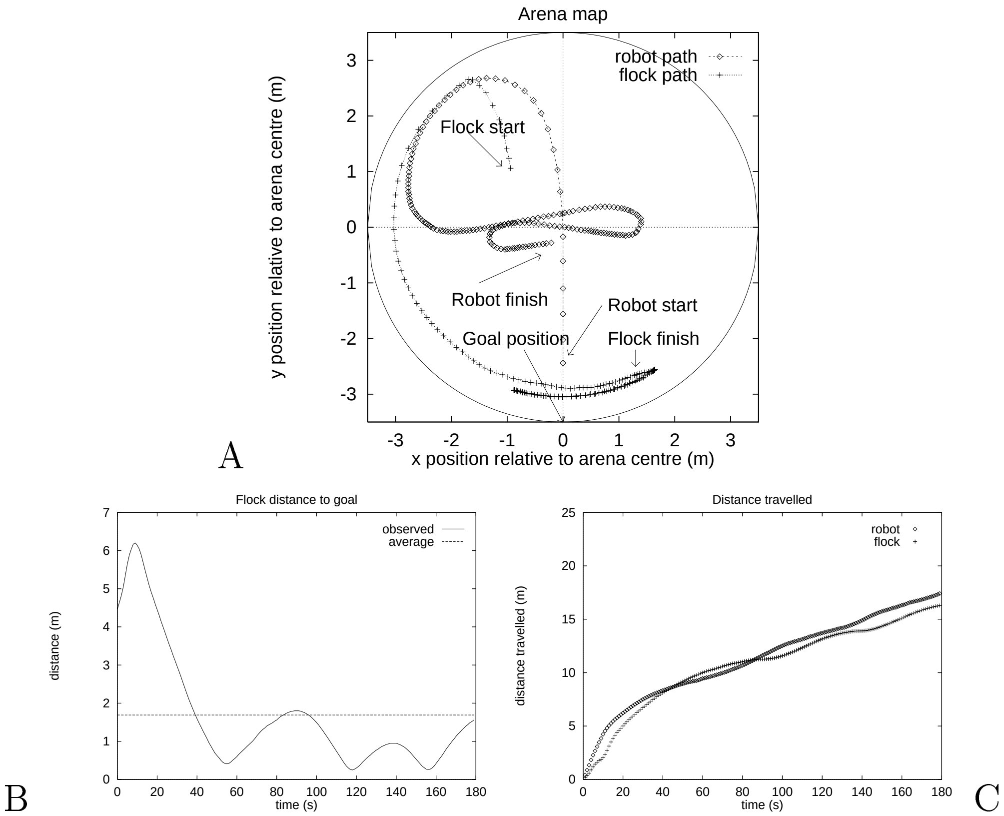


Figure 6.19: On-board rangefinder simulation results

stands off and begins the cyclic holding behaviour typical of Method 1 experiments (see Chapter 4). The flock is maintained near the goal, and oscillates about it. This is visible in the success (flock-to-goal distance) plot (B). The efficiency plot (C) shows the distances traveled by the robot and the flock. The last two screenshots of Figure 6.18 (8 & 9) show the flock being held around the goal.

These results show that local range sensing of the flock and walls can be used to achieve qualitatively similar behaviour to that produced by the earlier experiments which relied on global, absolute position measurements. An algorithm was presented which is very similar to that used in the earlier experiments, using simple processing of range data to generate a movement vector for the robot. However, this example required that a separate unexplained system provided information about the direction of the goal.

An immediate solution to this requirement might be to exploit the fact that the robot starts out at the goal and with a known heading. Odometry could be implemented to maintain an estimate of the robot's state which could be used to compute the direction to the goal. The inevitable odometry drift would mean that this position estimate would rapidly deteriorate unless it was corrected by combining it with some non-drifting (but possibly low-frequency) sensor data, such as a compass. The Kalman filter has become a standard technique to achieve this sensor fusion [Barshan and Durrant-Whyte, 1995, Leonard and Durrant-Whyte, 1991, Krotkov and Fuke, 1996, Kao, 1991, Roumeliotis et al., 1999]. In these circular arena experiments, the robot can accurately compute its distance from the center of the arena; this could provide another non-drifting input into the Kalman filter. In this way it is likely that a reasonable position estimation could be maintained for the length of a run (180 seconds in these experiments). However, this scheme exploits the particular constraints of this scenario (circular arena, walls always within sensor range, short duration trial, etc.) and will quickly fail as these constraints are relaxed (options for outdoor localization are briefly discussed in 7.4.1).

To work in the general case, the range-based controller needs a supplementary sensor system to reliably detect angle-to-goal, such as might be provided by a dedicated vision system. Even in the constrained version described here, multiple sensors (ranger finder, odometry, compass) and complex processing (Kalman filter) are required.

These requirements are eliminated in the next example, in which the same task is performed using only local vision data.

6.8.2 Vision method

This example demonstrates the application of the methods developed in this thesis to a robot with on-board vision. A simulated image is generated and supplied as the sole input to the robot's controller. The image is supplied segmented into ground, wall, ducklet and goal regions, indicated by green, blue, white and red pixels respectively (though they are reproduced as shades of grey on these pages). It is recognized that segmenting real images accurately and at high speed on a moving robot is a technical challenge which is out of the the scope of this project. Assuming the segmentation is possible, this method should transfer successfully into the real world.

Typical video cameras and lenses have a field of view of $< 90^o$, but this application requires a 360^o view. This requirement can be met using a conical mirror mounted above an upward-pointing camera; an arrangement used by Columbia University's 'Omnicamera' system

Figure 6.20: A Khepera robot fitted with a small CCD camera and conical mirror. This arrangement provides a 360^o view in the image projected onto the camera.

(http://www.cs.columbia.edu/CAVE/omnicam/) . A similar system is commercially available as 'CycloVision' (http://www.cyclovision.com/). Figure 6.20 shows a photograph of a Khepera robot in this configuration. The image on the conical mirror can be straightforwardly mapped into a conventional rectangular image, but some distortion due to an imperfect mirror, camera optics and resampling noise is inevitable.

Generating the sensor data

The robot is provided with a character array of 256x32 pixels. Each pixel has a value of either FLOOR, WALL, DUCKLET or GOAL, representing the corresponding feature in the environment. These features are projected into the array by the following method.

The array is initialized by filling it with FLOOR pixels (light grey in the figures). Then the range to nearest object is calculated along a ray for each of the 256 columns in the image, just as for the rangefinder example above. Rays are evenly spaced around the 360^o field of view, giving $360/256 = 1.4^o$ per column.

For each ray that detects a wall the image column is filled from the top down with WALL pixels (dark grey in the figures). The number of wall pixels filled in is inversely proportional to the square of the range-to-wall along that ray, scaled so that distant walls appear only at the top of the column, while nearby walls fill the column almost completely. Using only a few (32) vertical pixels gives a fairly course resolution which decreases rapidly with distance, but this is found to

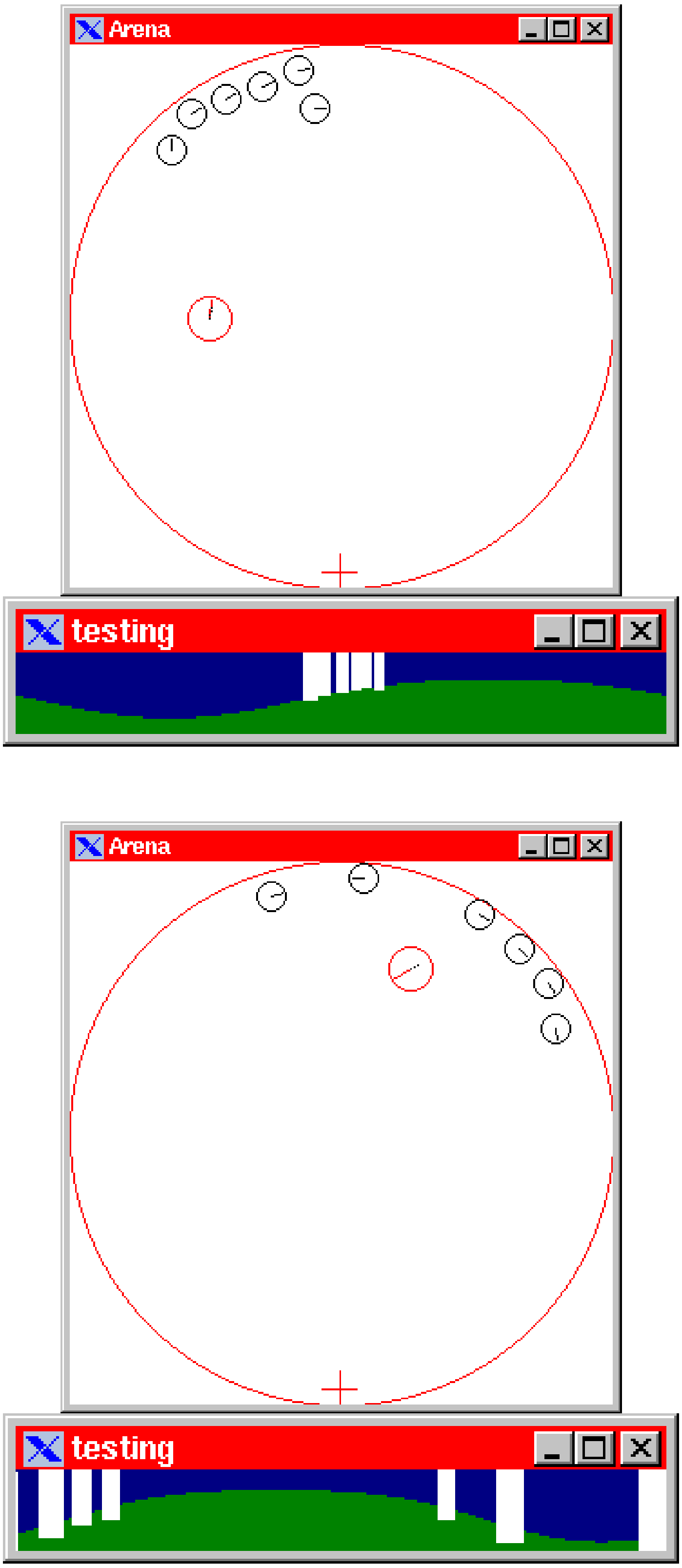


Figure 6.21: Screenshots from DuckSim showing overhead and robot's-eye views of the arena for two example scenarios. Floor is light grey, walls dark grey, ducks white.

be sufficient and allows very fast processing. The column that corresponds most nearly to the goal direction is filled in with GOAL pixels in place of WALL pixels. This simulates a marking on the wall which can be identified by the assumed segmentation algorithm. Ideally this could be from some source visible only to the robot such as an infra-red beacon, so that the ducks could not identify the goal themselves.

Similarly, for each ray that intersects a ducklet, the image column is filled from the top down with DUCKLET pixels (white in the figures).

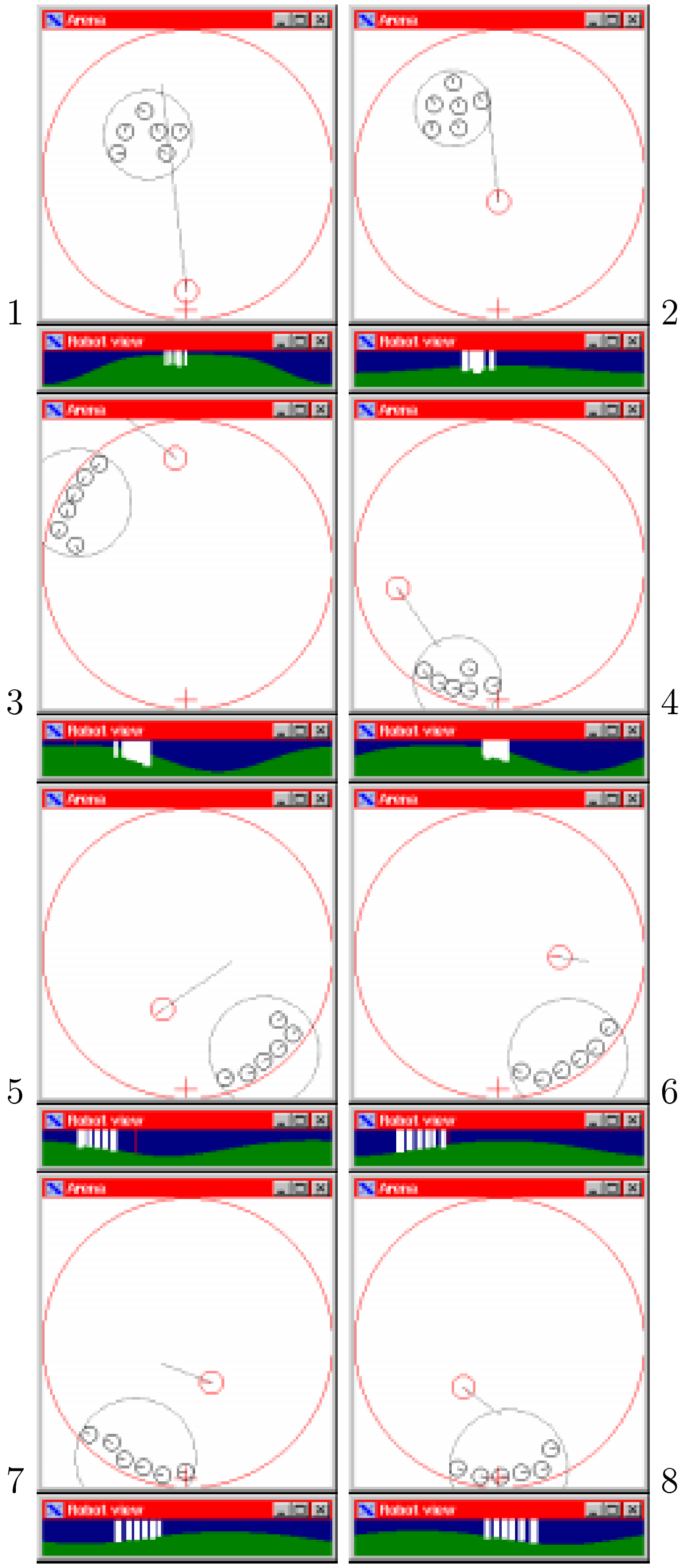
Once the array is filled in this way an image of the arena from the robot's point of view has been constructed. Figure 6.21 gives two examples of the 360^o view of the simulated arena generated by this method. The top example shows the robot approaching a tightly-clustered flock which appears as a small group of white bars in the synthetic image. The bottom example shows the robot splitting up the flock and facing away from it. In this case the ducklets are closer to the robot so the white bars are larger, and they are mostly behind it, so the bars are mostly at the edges of the image.

Controller

The controller used in this experiment is the most simple so far. Because the number of pixels of each object type varies according to the inverse-square of the objects' distance, a useful flock-control algorithm can be produced by simply summing the vector contribution of each pixel, where the type (colour) of the pixel determines its vector magnitude and its column number determines its vector angle: the robot is attracted to each of the DUCKLET pixels by a constant amount and repelled from WALL pixels by a constant amount. A repulsion of constant magnitude is required from the goal position, so the robot is repelled by a constant amount from the direction of the resultant of the GOAL pixel vectors. A record of the last detected direction of the goal is kept as a useful approximation in case the goal is obscured in the image by a ducklet.

Results

Figure 6.22 shows an example run with the visually guided robot as a sequential series of screenshots from the simulation. The small rectangles show the synthesized images generated from the arena in each case. Using just these images as input, the robot succeeds in approaching the ducklets (1,



Key: = robot, = ducklets, + = flock goal. Small windows show robot's eye-view of arena used as sole input to robot controller. Center of window corresponds to robot heading.

Figure 6.22: Sequence of images from the simulator during a on-board camera trial, showing successful behaviour.

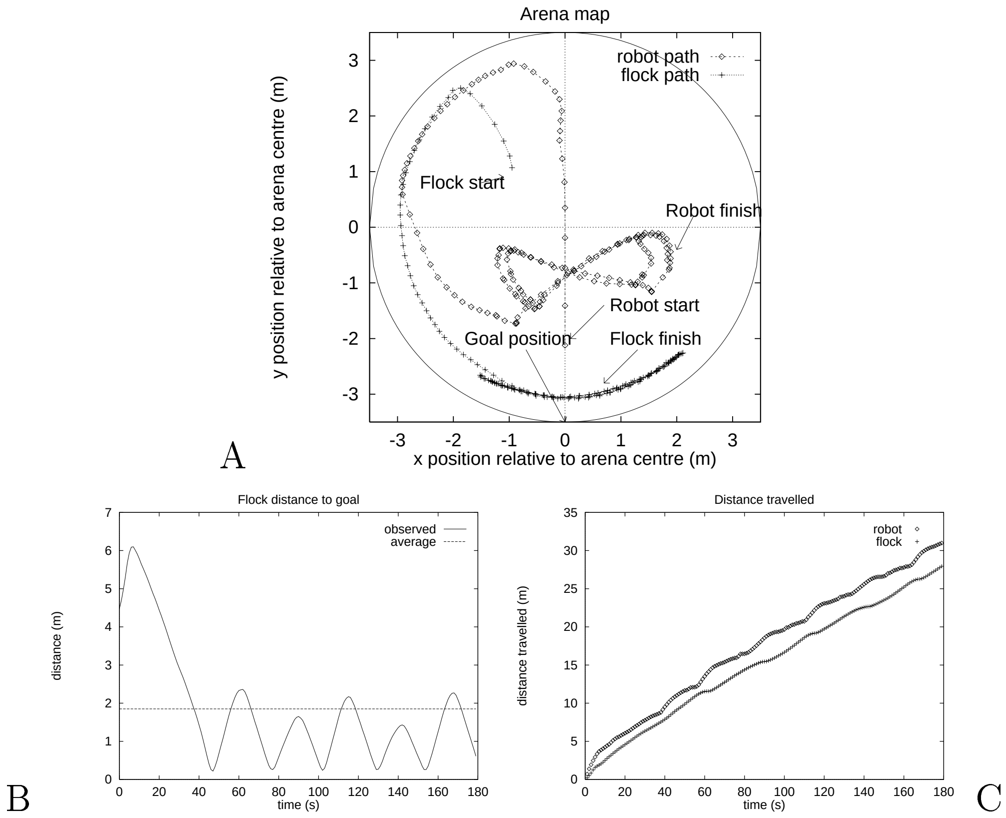


Figure 6.23: On-board camera simulation results

2), moving around behind them (3), pushing them towards the goal (4) and holding them in the oscillatory pattern typical of Method 1 and rangefinder experiments (4 to 8).

Figure 6.23 presents the results from the same trial, where (A) shows the paths taken by the robot and flock, (B) the success graph and (C) the efficiency graph. The undesirable oscillations of the flock about the goal and the robot about the flock can clearly be seen in (A) and (B). Methods of reducing this oscillation are discussed earlier in the thesis; in particular Method 2 succeeds by using a strategy whereby the flock is attracted to the flock by an amount proportional to the flock's distance to the goal. This is not directly observable by the rangefinder or vision methods, though can be calculated by triangulation given their respective positions relative to the robot. This could be a useful extension given the superiority of Method 2 over these Method 1-type local strategies, but has not been attempted here.

Chapter 7

Conclusion

7.1 Hypotheses vs. conclusions

Comparing the original hypotheses to the conclusions drawn from previous chapters:

1. *Robotic flock control can be achieved by exploiting the flock animals' threat-avoidance behaviour.*

   It was found that a group of ducks will move away from the robot vehicle described in Chapter 3 in a predictable manner. Controlling the position of the robot relative to the flock permits control of the direction and speed of movement of the flock. Robot controllers were constructed to exploit this behaviour; reproducing the basic ability of the sheepdog to drive a flock.

2. *The appropriate interaction is to position the robot behind the flock with respect to the goal while maintaining an appropriate robot-flock distance.*

   Chapters 4 and 5 described two robot controllers that position the robot in this way. A series of experiments showed that the flock is moved to the goal position in a majority of trials.

3. *A simulated flock could be used to design and test a robot controller that achieves (2).*

   The novel robot controller described in Chapter 4 was designed through consideration of, and experiment with, a simple simulated flock presented in Chapter 3. The controller was tested in simulation and found to achieve the goal task. The same controller was then implemented on the real robot and tested with flocks of ducks. The robot succeeded in gathering the flock in some trials, with a pattern of behaviour qualitatively similar to that of the simulation.

However, the controller was neither as successful nor as reliable in then real world as it was in simulation.

Chapter 5 presented a second novel robot controller which proved to be more successful in simulation than its predecessor. Transferring this controller to the real world robot, it was found that it was more successful and much more reliable than the original method, and that the behaviour of the real-world system approached that of the simulation more closely.

7.2 Summary of contributions

This thesis describes these novel contributions to the field:

1. demonstration of the first robot interaction with live animals to achieve a useful task;
2. development of two generic flock-control algorithms; including demonstration of them controlling both simulated and real flocks, and their application to three different sensor modalities.
3. development of a methodology for experiments in animal-interactive robotics without animals in the development cycle;
4. description of an appropriate vehicle and control architecture for interaction with a flock of ducks.

7.3 Summary of major design decisions

1. Domestic ducks as experimental animal

   Ducks were chosen because they show strong flocking behaviour and are small and relatively slow-moving. This allowed the interacting robot to be constructed on a manageable scale and cost. One of the studentships was in animal behaviour (Henderson) and SRI's area of study is agriculture. This dictated the choice of a mass-produced domestic species.

2. Circular 7m arena

   The largest available space at SRI was a workshop a little over 7m across. It was decided early on that an outside arena would pose significant problems for vision-based sensing due to lighting variation, so this was ruled out. The use of vision as the primary sensor was mandated by the original project proposal which allocated one of the studentships (Sumpter) on a computer vision topic.

A circle was chosen to avoid corners and thus minimize the complexity of the task. The team decided that a circle offered the higest chance of getting a real system working within the three year span of the project.

3. Circular, differential-drive robot

   Holonomic control is the simplest, most common mode of movement for mobile robots and is very common in the literature. This configuration offered maximum manoeverability and simpllicity of mechanical and control design.

4. Plain robot appearance

   The original intention was that Henderson's work would lead to insights about the visual design of a robot for flock control. The robot was originally designed with as null an appearance as possible; a blank grey cylinder. It quickly became apparent that the ducks would move away from this object reliably, and that this was sufficient to achieve the task. Henderson found that the plain cylinder was the least aversive stimulus when compared to a human and model fox, so that being herded by the robot rather than a human or dog could potentially offer welfare benefits [Henderson, 1999].

5. Off-board vision

   In the judgement of the project team, a static camera mounted above the arena offered the maximum likelihood of a successful system in the life of the project. However, the control methods devised do not totally rely on this arrangement for their success; this thesis has discussed suitable alternatives.

   The project required high-frequency image processing plus a fast, manoeverable robot, all at low cost. At the time of design (1995-6) the cost and size of a sufficiently poweful portable computer was prohibitive. A conventional workstation was chosen instead, connected to the robot by modem at the point which required lowest bandwidth communication.

6. Off-board control

At design time, it was not known how much computation would be required to run the eventual controller, so it was decided to be cautious and put the main controller off-board, too. The final controllers turned out to be very simple and could easily run on the robot's onboard computer.

7. Experiment with flock model first

   The behaviour of the flock is apparently complex, yet good flock models existed in the literature. It was hoped, and found, that building a simple flock model would provide insight into the interaction required to control the flock. Many weeks of experimentation with the model lead to the design of the Method 1 controller, which was constructed from the same conceptual parts as the model itself. This avoided any need for extensive experiments with the animals to rule out many prototype controllers.

7.4 Future research

7.4.1 Robot Sheepdog extensions

Many extensions and variations could be made to the task and strategies presented in this thesis. The issues of complex arena shapes, multiple robots and on-board sensing were discussed in Chapter 6, but three more extensions can be considered that might improve performance in the current task, or expand the task domain:

Unused information

Neither the original Method 1 or 2 uses the flock radius obtained by the vision system. This is a potentially useful piece of information that could be used to make the robot more responsive to flock state. The main reason for an increase in flock size would be when the flock splits into two or more sub-flocks, or when an individual breaks from the main flock. Flock animals strive to maintain proximity, so a separation can be considered a stressful event that should be avoided. Monitoring the size of the flock could allow the robot to back away when a split happened, allowing the animals to re-form a single flock.

Conversely, a split may be desirable, for instance if the task were to isolate individuals for inspection or veterinary attention. A sudden increase in flock size could indicate to a more sophisticated vision system that a single flock should be re-assessed as multiple sub-flocks or individuals.

The speed of the flock, though easily obtained from the existing tracking data, is not exploited in this thesis. A more sophisticated flock control strategy could be based on the relative *velocities* of the robot and flock, rather than the *positions* used up to now. This could perhaps allow more subtle control to further reduce animal stress or to improve absolute performance.

Adaptation for improved performance

This work has approached the issue of variation between flocks by designing a robust, general flock-control method. While it is successful in a majority of trials, its success and efficiency varies between flocks and over time with the same flock. While it is a stated requirement of an AIR system that it should not require manual optimization to work with any specific animal or group, the ability to self-optimize or *adapt* during run-time could be a very useful extension. Enhancing a general strategy with adaptation might allow a robot to immediately interact with a novel target flock in approximately the right way, with its performance improving over time as its experience increases.

Outdoor localization: Rover unleashed

The flock control methods described in this thesis rely on reliable information about the relative positions of the robot, flock and goal position. The localization schemes so far described so far have been designed for indoor experiments. If the Robot Sheepdog is to progress to more realistic environments, an outdoor localization scheme must be devised.

The *Global Positioning System* (GPS) provides a general solution to the outdoor localization problem, subject to a few caveats: (1) the receiving antenna must have line-of-sight to several transmitting satellites (nearby trees and buildings can attenuate signals enough to defeat GPS); (2) high resolution ($\pm$2cm) localization requires comparing the signals of two antennae, one at a 'base station' with fixed, known location for *Differential GPS* (DGPS); (3) GPS signals can drop out without warning for an unpredictable length of time; (4) (D)GPS has a relatively low-frequency update ($\approx$ 1Hz).

These limitations mean that GPS-fitted robots are typically equipped with high-frequency, high-availability instruments such as odometry and inertial sensors. The drift in these high-frequency sensors is compensated for by comparison with the GPS data when it is available. A Kalman filter can be employed for the sensor fusion (see references in section 6.8.1 above). Using this technique the University of Southern California has demonstrated two autonomous Pioneer robots moving in formation with a human-piloted helicopter [Montgomery and Sukhatme, 1999], all instrumented with GPS receivers.

This arrangement could provide a basis for an outdoor Robot Sheepdog system in which the robot, goal position, and animals are all located with by GPS with accuracy sufficient for the task. At the time of writing this may be impractical due to the high cost of GPS units, but better and cheaper systems are being produced every year.

7.5 Hot AIR

It is concluded that a behavioural simulation of animal behaviour can be sufficient to design a robot that can usefully interact with the real animal(s).

The design of an Animal-Interactive Robotic system must cope with the inevitable variation in behaviour between different animals and in the same animals over time. Therefore the designer can only exploit the basic, underlying principles; the 'core' behaviour, and not the specifics of any one animal, flock or trial. A perfect simulation of any animal is beyond the state of the art and is not available to the AIR designer. But given the robustness and flexibility demanded of any such system, a simple generic simulation model can be a useful design tool. If the model contains the complete core behaviour and no more, then a robot that interacts successfully with the model can be expected to interact successfully with the real animal. In addition, the act of constructing the model can inform the design of a subsequent robot controller.

As more and better animat models of animal behaviour emerge, whether generated by biologists to better understand the animal, or by AI scientists to better understand intelligent behaviour, the opportunities for the AIR designer increase. By allowing off-line experiments with these virtual animals, we may discover more ways of manipulating the real animals' environment to improve their welfare, or to do useful work, or both.

Appendix A

Improved flock tracking

A.1 Problem

This tracker suffered a problem which reduced its reliablity. The problem was peculiar to the application: the ducks continually created new high-contrast blobs of urine and faeces which the tracker would interpret as duck-pixels. Figure A.1 shows the tracker incorrectly incorporating faeces into the flock.

A.2 Solution

The first approach to this problem was to change the way the ducks were discriminated from the background. The ducks used were largely white, and the faeces were dark in shade, so instead of labelling pixels that were very *different* to the background, the pixels that were *lighter* than the background were chosen. This was found to actually reduce the performance of the tracker for two reasons: (1) the tracker no longer picked up the shadows of the ducks - reducing the number of useful duck-related pixels identified, and (2) the wetness of the new blobs made them shine under the lights so that they contained pixels which were lighter than the ducks themselves.

A second simple strategy was tried which worked very well. This time instead of examining every pixel in the flock-region, only every fifth pixel in each direction was examined. This reduced resolution tends to pick out the larger blobs (ducks) and not the smaller blobs. If the sparse sampling was done relative to the image, then it would be possible to find a faecal blob on a sampling point, causing it to be incorrectly identified. However, with the sampling grid measured relative to the flock centre a faecal blob may be picked up in one frame but due to the constant movement of the flock centre it is very unlikely that the blob will be detected in subsequent frames.

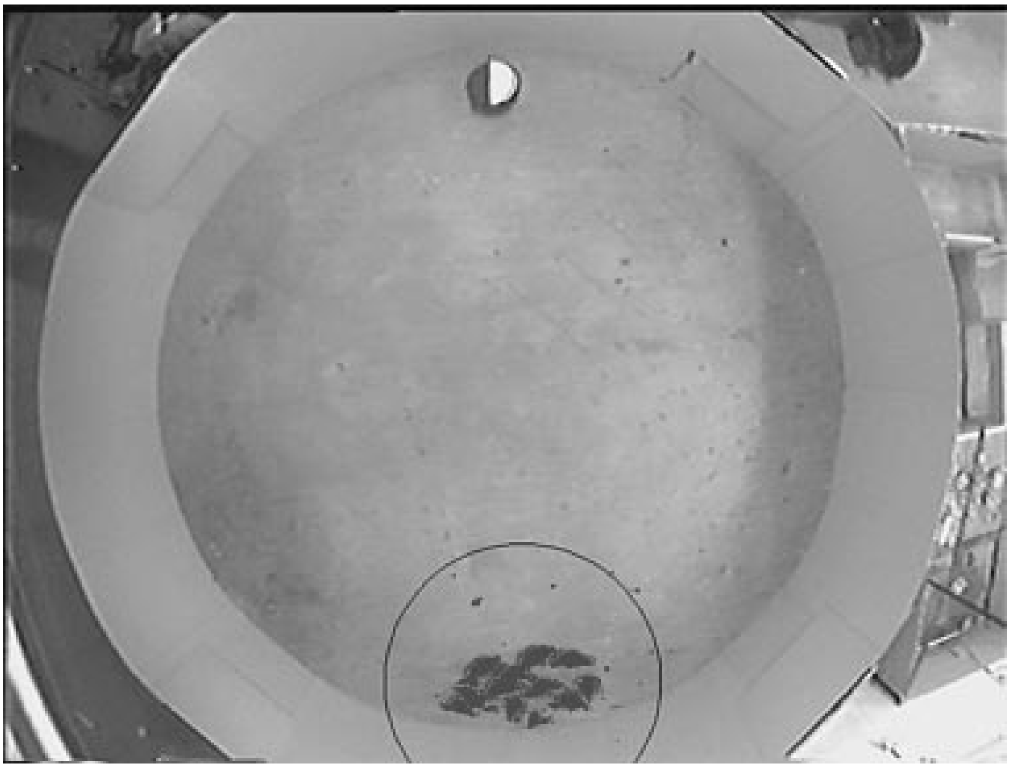


Figure A.1: The tracker used in the experiments occasionally incorporates duck excrement into the flock.

Thus over time the faecal blobs make little contribution to the detected flock shape. Figure A.2 shows two examples of the sparse-sampling tracker successfully locating the flock.

The sparse sampling strategy has a very useful side effect. The number of pixels examined goes down by a factor of 25, greatly increasing the overall speed. The current tracking system runs at over 30Hz - faster than the standard 25Hz video frame rate.

Figure A.2: Example images showing the improved tracker correctly locating the flock. The sparse sampling strategy does not pick up the duck excrement.

Appendix B

Henderson's thesis

Many references are made in the text to Jane Henderson's thesis. That study is complementary to this work and was produced simultaneously as part of the Robot Sheepdog Project. Her thesis abstract is reproduced here as an indication of the scope and goals of her work.

Flocking behaviour of ducks in response to predator stimuli

Abstract of Ph.D. thesis submitted to the University of Bristol, 1999

Jane V. Henderson

The visual stimuli used by domestic ducks for the assessment of enemies, and the responses of duck flocks to potential enemies were investigated. A technique was devised to measure a duckling's behavioural responses to a mobile enemy.

Ducklings appeared to use different visual stimuli to assess the threat posed by an approaching stimulus. Fear increased as stimuli approached, rising dramatically once the stimuli were within 4 m. When exposed to either a human, a taxidermist's model fox, or a vertical cylinder, the highest duckling responses occurred with the human, and the lowest with the cylinder. A model fox elicited higher fear responses than either a life-size photograph, or a fragmented photograph of the same fox, at distances less than 3.5m; the fragmented photograph elicited lower fear responses than the complete photograph at most distances. An increased total number of facial features was assessed by ducklings as being indicative of greater likelihood of attack. A conceptual model was proposed and validated of how different visual stimuli are used for enemy assessment as the stimulus-to-duckling distance varies.

Experiments to investigate flock dynamics found that individual ducklings maintained a relatively stable position within a flock whilst being herded. Individual fearfulness was found to be a predictor of position in a flock, when fearfulness was measured in the same context as flock herding. Some differences in flock response were found when flock composition was varied according to attributes of individual ducklings.

Based on these findings, guidelines were produced for the design of robots to work amongst animals. The most suitable design will depend upon the task the robot has to perform, and its frequency.

Bibliography

[Agre and Chapman, 1990] Agre, P. E. and Chapman, D. (1990). What are plans for? In [Maes, 1990], pages 17–34.

[Anderson and Donath, 1990] Anderson, T. L. and Donath, M. (1990). Animal behavior as a paradigm for developing robot autonomy. *Robotics and Autonomous Systems*, 6:145–168.

[Arai et al., 1996] Arai, F., Tanimoto, M., Fukuda, T., Shimojima, K., Matsuura, H., and Negoro, M. (1996). Multimedia tele-surgery using high speed optical fiber network and its application to intervascular neurosurgery. In [Caplan and Lee, 1996].

[Barshan and Durrant-Whyte, 1995] Barshan, B. and Durrant-Whyte, H. F. (1995). Inertial navigation systems for mobile robots. *IEEE Transactions on Robotics and Automation*, 11(3):328–342.

[Boden, 1987] Boden, M. A. (1987). *Artificial Intelligence and Natural Man.* MIT Press, Cambridge, MA., 2 edition.

[Boekhorst and Hogeweg, 1994] Boekhorst, I. and Hogeweg, P. (1994). Effects of tree size on travelband formation in orang-utans: Data analysis suggested by a model study. In Brooks, R. A. and Maes, P., editors, *Procedings of the Fourth International Conference on Artificial Life*, pages 119–129, Cambridge, Mass. MIT Press.

[Boyle et al., 1993] Boyle, R., Hlavac, V., and Sonka, M. (1993). *Image Processing, Analysis and Machine Vision.* Chapman and Hall.

[Braitenberg, 1984] Braitenberg, V. (1984). *Vehicles.* MIT Press, Cambridge MA.

[Brooks, 1986] Brooks, R. A. (1986). A robust layered control system for a mobile robot. In *IEEE Journal of Robotics and Automation*, volume RA-2 (1), pages 14–23.

[Brooks, 1991] Brooks, R. A. (1991). Intelligence without reason. A.I. memo no. 1293, MIT Artificial Intelligence Laboratory.

[Caldwell, 1986] Caldwell, G. (1986). Predation as a selective force on foraging herons: Effects of plumage color and flocking. *The Auk*, 103:494–505.

[Cameron and Probert, 1994] Cameron, S. and Probert, P. (1994). *Advanced Guided Vehicles: Aspects of the Oxford AGV project.* World Scientific.

[Caplan and Lee, 1996] Caplan, N. and Lee, C. S. G., editors (1996). *IEEE International Conference on Robotics and Automation.*

[Cliff and Bullock, 1993] Cliff, D. and Bullock, S. (1993). Adding "foveal vision" to Wilson's animat. *Adaptive Behaviour*, 2(1):49–72.

[Cliff et al., 1994] Cliff, D., Husbands, P., Meyer, J., and Wilson, S., editors (1994). *From Animals to Animats 3: Proceedings of the third international conference on the simulation of adaptive behaviour*, Cambridge, Mass. MIT Press.

[Connell, 1990] Connell, J. H. (1990). *Minimalist Mobile Robotics.* Academic Press, Boston.

[Cresswell, 1994] Cresswell, W. (1994). Flocking is an effective anti-predation strategy in redshanks. *Animal Behaviour*, 47:433–442.

[Duncan et al., 1987] Duncan, I., Slee, G., Kettlewell, P., Berry, P., and Carlisle, A. (1987). Comparison of the stressfulness of harvesting broiler chickens be machine and by hand. *B. Poultry Science*, 27:109–114.

[Frost et al., 1993] Frost, A., Street, M., and Hall, R. (1993). The development of a pneumatic robot for attaching a milking machine to a cow. *Mechatronics*, 3(3):409–418.

[Grandin, 1989] Grandin, T. (1989). Behavioural principles of livestock handling. *Professional Animal Scientist*, pages 1–11.

[Hague et al., 1990] Hague, T., Cameron, S., and Brady, M. (1990). Using moments to plan paths for the Oxford AGV. In *Proceedings of the IEEE Conf. Robotics and Automation*, Cincinatti.

[Hallam and Hallam, 1994] Hallam, B. and Hallam, John C. T. amd Halperin, J. R. P. (1994). An ethological model for implementation in mobile robots. *Adaptive Behavior*, 3(1):51–79.

[Hallam and Hayes, 1994] Hallam, B. and Hayes, G. (1994). Comparing robot and animal behaviour. Research Paper 598, Department of Artificial Intelligence, University of Edinburgh.

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

[Henderson, 1997] Henderson, J. V. (1997). Adaptive responses of animals to mobile robots. Technical report, University of Bristol.

[Henderson, 1999] Henderson, J. V. (expected 1999). *Adaptive Responses of Herding Animals to a Robot Vehicle*. PhD thesis, University of Bristol.

[Jakobi, 1998] Jakobi, N. (1998). Evolutionary robotics and the radical envelope of noise hypothesis. Technical report, University of Sussex.

[Kao, 1991] Kao, W. (1991). Integration of gps and dead-reckoning systems. In *Proceedings of the IEEE Vehicle Navigation and Information Systems*, pages 635–643.

[Kleiner, 1994] Kleiner, K. (1994). Look to the insect. *New Scientist, No. 1951, 12 Nov. 1994*, 144:27–29.

[Koza, 1992] Koza, J. (1992). *Genetic Programming*. MIT Press, Cambridge, Mass.

[Krotkov and Fuke, 1996] Krotkov, E. and Fuke, Y. (1996). Dead reckoning for a lunar rover on uneven terrain. In *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pages 411–416.

[Leonard and Durrant-Whyte, 1991] Leonard, J. J. and Durrant-Whyte, H. F. (1991). Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3):376–382.

[Maes, 1990] Maes, P., editor (1990). *Designing Autonomous Agents*. The MIT Press: Cambridge, MA.

[Maes and Brooks, 1990] Maes, P. and Brooks, R. A. (1990). Learning to co-ordinate behaviours. In *AAAI-90, Proceedings of the 8th National Conference on Artificial Intelligence*, volume 2, pages 796–802.

[Maes et al., 1996] Maes, P., Mataric, M., Meyer, J.-A., Pollack, J., and Wilson, S., editors (1996). *From Animals to Animats 4: Proceedings of the third international conference on the simulation of adaptive behaviour*, Cambridge, Mass. MIT Press.

[Manning, 1972] Manning, A. (1972). *An Introduction to Animal Behaviour*. Edward Arnold, second edition.

[Martin, 1997] Martin, D. (1997). Watt poultry statistical yearbook. *Poultry International*, 35(8).

[Martin and Bateson, 1993a] Martin, P. and Bateson, P. (1993a). *Measuring Behaviour.* Cambridge University Press.

[Martin and Bateson, 1993b] Martin, P. and Bateson, P. (1993b). *Measuring Behaviour.* Cambridge University Press.

[Mataric, 1992] Mataric, M. (1992). Minimizing complexity in controlling a mobile robot poulation. In *Proc. IEEE International Conference on Robotics and Automantion*, pages 830–835.

[Mataric, 1994] Mataric, M. J. (1994). Learning to behave socially. In [Cliff et al., 1994], pages 453–462.

[Mataric and Cliff, 1996] Mataric, M. J. and Cliff, D. (1996). Challenges in evolving controllers for physical robots. *Journal of Robotics and Autonomous Systems*, 19(1):67–83.

[Maynard-Smith, 1982] Maynard-Smith, J. (1982). *Evolution and the theory of games.* Cambridge University Press.

[McFarland, 1985] McFarland, D. (1985). *Animal Behaviour.* Longman Scientific & Technical.

[McFarland and Bosser, 1993] McFarland, D. and Bosser, T. (1993). *Intelligent behavior in animals and robots.* MIT Press.

[Miller, 1994] Miller, G. F. (1994). Artificial life as theoretical biology: How to do real science with computer simulation. Technical report, School of Cognitive and Computing Sciences, University of Sussex.

[Miller and Cliff, 1994] Miller, G. F. and Cliff, D. (1994). Co-evolution of pursuit and evasion I: Biological and game-theoretic foundations. Technical Report CSRP311, School of Cognitive and Computing Sciences, University of Sussex.

[Montgomery and Sukhatme, 1999] Montgomery, J. and Sukhatme, G. (1999). Avatar autonomous helicopter project. unpublished work in progress at the University of Southern California, http://www-robotics.usc.edu/ avatar.

[Nepomnyashchikh and Gremyatchikh, 1996] Nepomnyashchikh, V. and Gremyatchikh, V. (1996). The experimental study and computer simulation of fish behaviour in the uniform environment. In [Maes et al., 1996].

[Parrish, 1992] Parrish, J. (1992). Do predators shape fish schools: interactions between predators and their schooling prey. *Netherlands Journal of Zoology*, (42):358–370.

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

[Pitcher and Parrish, 1993] Pitcher, T. and Parrish, J. (1993). Functions of shoaling behaviour in teleosts. In Pitcher, T., editor, *Behaviour of Teleost Fishes.* Chapman and Hall, second edition.

[Prescott, 1995] Prescott, N. B. (1995). *Dairy cow behaviour and automatic milking.* PhD thesis, University of Bristol.

[Ranta et al., 1993] Ranta, E., Hannu, R., and Lindstrom, K. (1993). Competition versus cooperation: success of individuals foraging alone and in groups. *The American Naturalist*, (142):42–58.

[Renault et al., 1990] Renault, O., Thalmann, N., and Thalmann, D. (1990). A vision based approach to behavioral animation. *The Journal of Visualisation and Computer Graphics*, 1:18–21.

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

[Reynolds, 1992] Reynolds, C. W. (1992). An evolved, vision-based behavioral model of coordinated group motion. In Meyer, J. and Wilson, S., editors, *From Animals to Animats (Proceedings of Simulation of Adaptive Behaviour)*. MIT Press.

[Reynolds, 1994] Reynolds, C. W. (1994). An evolved, vision-based behavioral model of obstacle avoidance behaviour. In Langton, C. G., editor, *Artificial Life III*, volume XVII of *SFI Studies in the Sciences of Complexity*, pages 327–346. Addison-Wesley, Santa Fe Institute, New Mexico, USA.

[Roumeliotis et al., 1999] Roumeliotis, S., Sukhatme, G., and Bekey, G. (1999). Circumventing dynamic modeling: Evaluation of the error-state kalman filter applied to mobile robot localization. In *Proceedings of the 1999 IEEE International Conference in Robotics and Automation*.

[Ryer and Olla, 1991] Ryer, C. and Olla, B. (1991). Information transfer and the facilitation and inhibition of feeding in a schooling fish. *Environmental Biology of Fishes*, (30):317–323.

[Shaw, 1970] Shaw, E. (1970). Schooling in fishes: critique and review. In Tabach, editor, *The development and evolution of behaviour*. WH Freeman and Co., San Francisco.

[Siegel, 1956] Siegel, S. (1956). *Nonparametric statistics for the behavioural sciences*. McGraw-Hill.

[Spier and McFarland, 1996] Spier, E. and McFarland, D. (1996). A finer-grained motivational model of behaviour sequencing. In [Maes et al., 1996].

[Sumpter, 1999] Sumpter, N. (expected 1999). *Robotic Sheepdog: Modelling Animal Behaviours from Image Sequences (preliminary title)*. PhD thesis, School of Computer Studies, University of Leeds.

[Sumpter et al., 1997] Sumpter, N., Boyle, R., and Tillett, R. (1997). Modelling collective animal behaviour using extended point distribution models. In *Proc. British Machine Vision Conference*, pages 242–251.

[Sumpter and Bulpitt, 1998] Sumpter, N. and Bulpitt, A. (1998). Learning spatio-temporal patterns for predicting object behaviour. Technical report, University of Leeds, School of Computer Studies, The University of Leeds, UK.

[Sumpter et al., 1998] Sumpter, N., Bulpitt, A., Vaughan, R., Tillet, R., and Boyle, R. (1998). Learning models of animal behaviour for a robotic sheepdog. In *Proceedings of the IAPR Workshop on Machine Vision Applications*.

[Trevelyan, 1989] Trevelyan, J. P. (1989). Sensing and control for sheep-shearing robots. *IEEE Journal of Robotics and Automation*, 5(6):716–727.

[Trevelyan, 1992] Trevelyan, J. P. (1992). *Robots for Shearing Sheep: Shear Magic*. Oxford University Press.

[Trevelyan et al., 1983] Trevelyan, J. P., Kovesi, P. D., and Ong, M. C. H. (1983). Motion control for a sheep shearing robot. In *Robotics Research, ed. Brady and Paul, MIT Press, 1984. (Proceedings of the First International Symposium on Robotics Research, Bretton Woods August, 1983.)*, pages 175–190.

[Tsai, 1987] Tsai, R. (1987). A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation*, RA-3(4):323–344.

[Vaughan et al., 1997] Vaughan, R., Henderson, J., and Sumpter, N. (1997). Introducing the robot sheepdog project. In Juste, F., Andreu, G., Valiente, J., and Benlloch, J., editors, *Proceedings of the International Workshop on Robotics and Automated Machinery for BioProductions*.

[Vaughan et al., 1998a] Vaughan, R., Sumpter, N., Frost, A., and Cameron, S. (1998a). Experiments in automatic flock control. In *Proceedings of the Sixth Symposium on Intelligent Robotic Systems (SIRS'98)*.

[Vaughan et al., 1998b] Vaughan, R., Sumpter, N., Frost, A., and Cameron, S. (1998b). Robot control of animal flocks. In *Proceedings of the Joint Conference on the Science and Technology of Intelligent Systems (ISIC/CIRA/ISAS'98)*. IEEE.

[Vaughan et al., 1998c] Vaughan, R., Sumpter, N., Frost, A., and Cameron, S. (1998c). Robot sheepdog project achieves automatic animal control. In *From Animals to Animats 5: Proceedings of the Fifth International Conference on the Simulation of Adaptive Behaviour*. MIT Press.

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

[Webb, 1993] Webb, B. (1993). *Perception in Real and Artificial Insects: Robotic Investigation of Cricket Phonotaxis*. PhD thesis, University of Edinburgh.

[Webb, 1994] Webb, B. (1994). Robotic experiments in cricket phonotaxis. In [Cliff et al., 1994], pages 45–54.

[Webb and Hallam, 1996] Webb, B. and Hallam, J. (1996). How to attract females: further robotic experiments in cricket phonotaxis. In [Maes et al., 1996], pages 75–83.

[Weisstein, 1998] Weisstein, E. W. (1998). *CRC Concise Encyclopedia of Mathematics*. CRC Press. restricted online use at http://astsun.astro.virginia.edu/ eww6n/math/math.html.

[Werner and Dyer, 1992] Werner, G. M. and Dyer, M. G. (1992). Evolution of herding behaviour in artificial animals. In Meyer, J. and Wilson, H. R. S., editors, *From Animals to Animats 2: Proceedings of the second international conference on the simulation of adaptive behaviour*, Cambridge, Mass. MIT Press.

[Wheeler and de Bourcier, 1994] Wheeler, M. and de Bourcier, P. (1994). What is synthetic behavioural ecology? In Lemmen, R. and Thompson, A., editors, *The Seventh White House Papers: Graduate Research into the Cognitive and Computing Sciences at Sussex. CSRP 350.* University of Sussex.

[Wilson, 1985] Wilson, S. W. (1985). Knowledge growth in an artificial animal. In Grefenstette, J. J., editor, *Proceedings of an International Conference on Genetic Algorithms and their applications (ICGA85), Pittsburgh PA.*, Hillsdale NJ. Lawrence Erlbaum Associates.

[Young, 1988] Young, D. (1988). Representing images for computer vision. Technical report, School of Cognitive and Computng Sciences, University of Sussex, CSRP.96.

[Yukawa et al., 1996] Yukawa, T., Khalid, M., Uchiyama, M., Inooka, H., Oshashi, T., Obinata, G., Kagaya, H., and Shimada, Y. (1996). Modular hybrid functional electrical stimulation system. In [Caplan and Lee, 1996].

[Zaera et al., 1996] Zaera, N., Cliff, D., and Bruten, J. (1996). (not) evolving collective behaviours in synthetic fish. In [Maes et al., 1996].