Dynamic Virtual Fences for Controlling Cows

Z. Butler[1], P. Corke[2], R. Peterson[1], and D. Rus[1,3]

[1] Dartmouth Computer Science Department
Hanover, NH 03755 USA
{zackb,rapjr,rus}@cs.dartmouth.edu
[2] CSIRO ICT Centre
Brisbane, Australia
peter.corke@csiro.au
[3] Computer Science and Artificial Intelligence Laboratory, MIT
Cambridge MA 02139, USA
rus@csail.mit.edu

Abstract. A virtual fence is created by applying an aversive stimulus to an animal when it approaches a predefined boundary. It is implemented by a small animal-borne computer system with a GPS receiver. This approach allows the implementation of virtual paddocks inside a normal physically fenced paddock. Since the fence lines are virtual they can be moved by programming to meet the needs of animal or land management. This approach enables us to consider animals as agents with natural mobility that are controllable and to apply a vast body of theory in motion planning. In this paper we describe a herd-animal simulator and physical experiments conducted on a small herd of 10 animals using a Smart Collar. The Smart Collar consists of a GPS, PDA, wireless networking and a sound amplifier. In particular we describe a motion planning algorithm that can move a virtual paddock which is suitable for mustering cows. We present simulation results and data from experiments with 8 cows equipped with Smart Collars.

1 Introduction

Our goal is to develop computational approaches for studying groups of agents with natural mobility and social interactions. Such systems differ in many ways from engineered mobile systems because their agents can move on their own due to complex natural behaviors as well as under the control of the environment (for example moving toward a food or water source). We wish to generate models of such systems using observed physical data and to use these models to synthesize controllers for the movement of the mobile agents. Our main motivation and application is in the agricultural domain. Herds of animals such as cattle are complex systems. There are interesting interactions between individuals, such as friendship, kinship, group formation, leading and following. There are complex interactions with the environment, such as looking for a water source in a new paddock by perimeter tracing along the fence and random walking within the perimeter. Such behaviors are well known to farmers but not so well documented. Unlike more familiar robot control problems, the animal state (stress, hunger, desire) is only partially observable and only limited control over motion can be exerted.

In this work we combine robotics, networking and animal behavior to create a fence-less approach to herding cows called *control by virtual fences*.

There are two fundamentally different approaches to controlling animal position: a physical agent such as a sheepdog or robot, and a stimulation device worn by the animal. In the first category there is the pioneering work of Vaughan[8] who demonstrated a mobile robot that was able to herd a flock of ducks to a desired location within a circular pen. In the second category there are a number of commercial products used to control domestic pets such as dogs. These typically employ a simple collar which provides an electric shock when it is in close proximity to a buried perimeter wire. The application of smart collars to manually control cattle is discussed in detail by Tiedemann and Quigley[7,5]. The idea of using GPS to automate the generation of stimuli is discussed in [4,2].

In [3] we describe our first experiments to controlling a herd of cows with a single static virtual fence using an approach that relies on ad-hoc networking. This paper extends [3] on the algorithmic side, by introducing motion planning for computing dynamic virtual fences whose goal is to muster the herd to a new location.

2 Approach

Our virtual fences combine GPS localization, wireless networking, and motion planning to create a fence-less approach to herding animals (see Figure 1). Each animal is fitted with a smart collar consisting of a GPS unit, a Zaurus PDA, wireless networking, and a sound amplifier. The collar is given the boundary of a polygonal virtual paddock in the form of a set of linear fences specified by their coordinates. The location of the animal is periodically checked against this polygon using the collar GPS. When in the neighborhood of a fence, the animal is given a sound stimulus whose volume is proportional to the distance from the boundary, designed to keep the animal within boundaries.

Each virtual fence is defined by a point F_p and a normal vector F_n. This representation allows for an easy computation of the distance d that the cow is behind (or in front of) the fence. If d is positive, the cow is in the desired region. Several fences can be combined to represent an enclosed boundary. We consider the virtual fence to be a repulsive potential field whose magnitude increases with distance beyond the fence line. Such a graduated stimulus will help the animal better understand the location of the fence [3,2]. The magnitude of the field is rendered in terms of sound stimulus volume or stimulus rate. A more sophisticated approach is to monitor d over time, and stop the stimulus as soon as the cow begins to move toward the desired region.

Cattle domain experts have suggested using a library of naturally occurring sounds that are scary to the animals (a roaring tiger, a barking dog, a hissing snake) and randomly rotating between the sounds.

A virtual fence can be made dynamic by automatically and gradually shifting its location. A moving fence can be instantiated with a non-zero velocity F_v, in m/s. The point F_p is then moved as a function of time along the normal, $F_p(t) =$

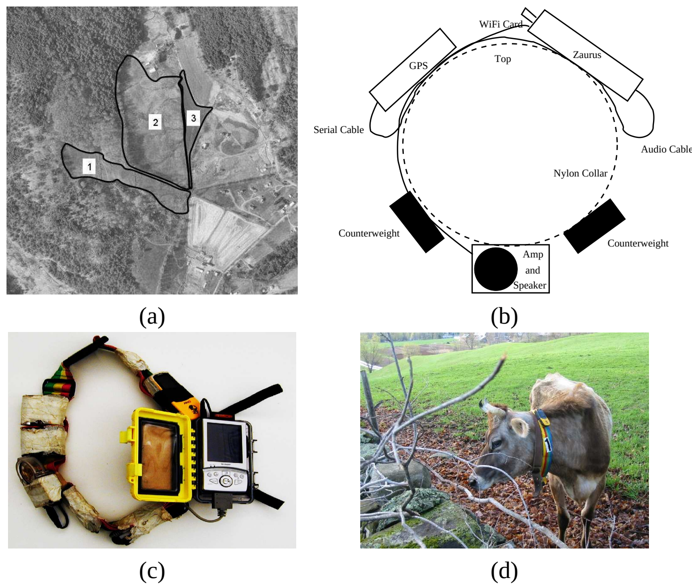


Fig. 1. (a) Aerial view of Cobb Hill farm. The fields where experiments were conducted are outlined in black. North is up. The photo displays an area approximately 1 km on a side. (b) The components of the Smart Collar include a Zaurus PDA, WiFi compact flash card, eTrex GPS, protective case for the Zaurus, an audio amplifier with speaker, and various connecting cables. (c) A fully assembled Smart Collar, with PDA case open. (d) A cow with a collar.

$F_p(0) + \gamma F_n F_v t$. Several moving fences can be used to muster the herd according to the plans developed by our motion planning algorithm, described in Section 3.

2.1 System hardware

Prototypes of a Smart Collar were constructed using commercial off-the-shelf components that are readily available. Figure 1(b) shows the components of a collar. The computer is a Zaurus PDA with a 206MHz Intel StrongArm processor, 64MB of RAM, with an additional 128MB SD memory card. It runs Embedix Linux with the Qtopia window manager. The Zaurus has a serial port and stereo sound port. A Socket brand 802.11 compact flash card provides a wireless network connection. An eTrex GPS unit is connected to the serial port of the Zaurus. A small Smokey brand guitar amplifier is used to reproduce sounds from the Zaurus audio port. A fully assembled collar is shown in Figure 1(c). Figure 1(d) shows a cow wearing an early version of the collar.

2.2 Software Infrastructure

The components of the software used in the experiments are as follows:

Fences and Sounds Fences can be added or removed at any time and several of them can be created at once from definitions stored in a file. Several fences can be combined to create convex polygonal shapes. When the GPS readings indicate a cow has crossed a fence a sound is triggered. The sounds are stored in WAV format files and can be selected from a list to be played on the Zaurus audio device. The volume of sounds is controllable on a percentage scale from zero to 100 percent. All fences use the currently selected sound and volume, which can be changed without redefining the fences.

The fence module also reads and interprets the GPS data which arrives every two seconds when the GPS has a good lock on the satellites. It also sends a periodic Alive message indicating the collar is functional.

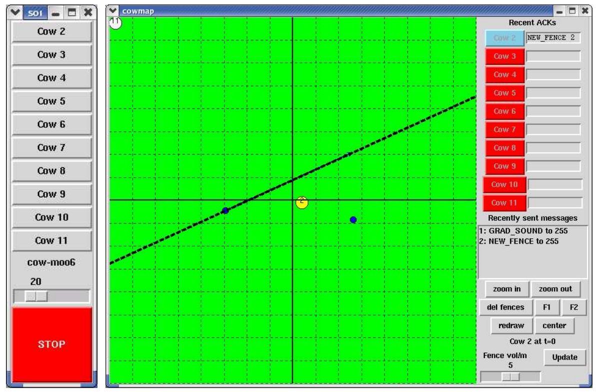
Message Handling We developed our own adhoc messaging protocol layered over 802.11 sockets. All WiFi messages are multihop, being forwarded once by each collar, to improve range and connectivity within the herd. There are two message channels, one outgoing from a basestation and one incoming to the basestation. The outgoing channel is used for defining fences, manually triggering sounds, setting sound type and volume. The incoming channel carries "Alive" messages indicating a collar is active, and acknowledgment messages for receipt and proper interpretation of messages. Figure 4(a) shows the number of hops required for an Alive message to reach the laptop basestation during an experiment. Most messages are relayed only once to reach their destination which indicates good connectivity between collars. Dynamic graphs of the message routing have shown us that connectivity among the herd is usually quite good since the cows tend to stay near each other. Connectivity with the base station was problematic in that there is a tradeoff in staying far enough away to not influence the herd. WiFi networks are essentially line of sight and are blocked completely at times by the cows bodies.

Experiment Control Both text and GUI control programs are used to manage the collars in the field. The text control program can be run on a Zaurus or Linux laptop and allows setting and deleting fences, setting type and volume of sound, and manually triggering a sound. The GUI control programs (see Figure 2) include the functionality of the text program, and add buttons for triggering sounds on specific cows, a map display showing current cow locations and status (i.e., relationship to fence boundary and whether a sound is playing), and a status display showing whether Alive messages have been received recently from each cow.

3 Planning for Dynamic Fences

An important goal of this work is to automatically muster cows from one pasture to another. To make this possible, we have started to develop a path planning system for virtual fences. While this problem shares some basic features with traditional robot path planning, it has important differences as well.

The planner creates a path from one point to another using a simple occupancy grid and A* search[6]. The object executing the plan is a virtual (polygonal) paddock,



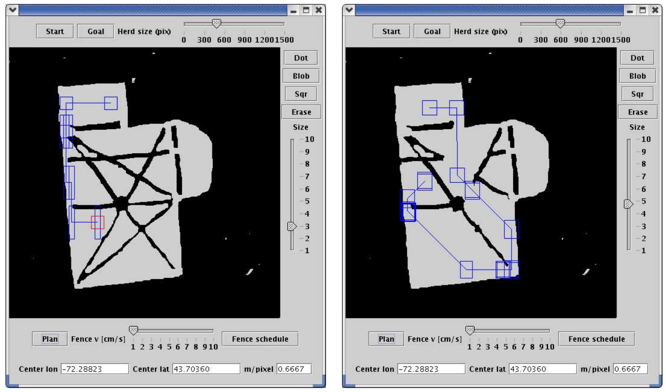
(a) (b)

Fig. 2. GUIs used on laptops to monitor field experiments. (a) Sound control GUI. Pressing a button triggers the current sound on a specific cow. Current sound and volume can also be selected. (b) Map control GUI. Shows the last reported position of each cow, whether it is currently playing a sound, and whether an Alive message has been received recently. Buttons and text boxes in upper right show recent command acknowledgments from collars.

with significant extent that may change as it moves[1], as long as its area remains sufficiently large for the herd. Thus, it is easier to perform planning in the workspace than the configuration space. Obstacles can overlap somewhat with the virtual fence edges, changing the effective area of the virtual paddock but not altering the plan. We use planning operators that change the dimensions of the paddock while keeping the amount of free space within the paddock sufficient for the given number of animals. Finally, we would like the motion to consist of a small number of straight-line segments. This type of optimization is necessary because changing the animals' direction is more difficult and confusing than keeping them moving along their current vector, and to limit the number of fences that are downloaded to the collars. This optimization can be implemented by giving turns a large cost in the A* search.

To create a plan, instead of doing an expensive search in five dimensions (paddock location, width, height and motion direction) we first generate a pair of baseline plans. The first uses a point-sized virtual paddock and the second a constant-sized square paddock. These plans can be computed using the motion operators only. If the two plans are similar in length and complexity then the square paddock plan can be used as is. If not, we then gradually reduce the size of the square paddock until a plan is successful. The successful plan is then used as a base for a path with a variable-sized virtual paddock. The search for this final path is done efficiently in three dimensions (width, height and distance along the base path.) The GUI developed for this planner along with examples of resulting plans is shown in Fig. 3.

[1] Both extent and shape may change.



(a) (b)

Fig. 3. GUI for dynamic fence planning. Environment is based on the Dartmouth Green, with some real-world paths defined to be obstacles. In (a), finding a path from the start (in top portion of free space) to the goal (at lower left) requires a significant change in the size and shape of the virtual paddock. (b) A more complex path — note that small overlap with obstacles is considered acceptable as long as sufficient free space remains in the virtual paddock.

Plans are turned into schedules of fences that are executable by the collar. For each segment of the path, four fences are required to define the paddock. A velocity is set for the fences, which in turn gives each segment a time interval over which its fences are active. This list of fences, with a point, normal vector, speed and relative time interval for each, is then given to the collars. The collars wait for an initialization message which tells them to make the time intervals absolute from that moment and continue to evaluate the fences and make appropriate stimuli for the course of the path.

4 Results

We have implemented both static and dynamic virtual fences in simulation and on 10 smart collars. Simulations were developed to investigate various algorithmic and stimulus methods. Early field experiments were performed on a small herd of cows at Cobb Hill Farm in Vermont. Later experiments were performed with people on the Dartmouth campus.

4.1 Simulation Experiments

To test the various virtual fence techniques, we developed a Matlab simulator that models the behavior of a herd of cows both with and without the virtual fence stimulus. We were inspired by Vaughan's duck simulator[8], but extended the animal model to account for the differences between the species as well as their environments. Most importantly, while we also use potential fields to model the effects of one animal's position on another's motion, we explicitly model the stress of each animal and use this to affect the animal's behavior. The animals have a two-state behavior model, walking and grazing, each with associated speeds and durations. In terms of motion, we use the potential force as a force on the cow, but model the cows as non-holonomic and give them a maximum angular velocity. If the virtual force given by the potential fields is not closely aligned with the cow's current direction, the cow will turn until the force causes it to walk in a reasonable direction.

In the simulation, stress is created by the fence stimulus as well as the nearby presence of other fast-moving animals or isolation from the herd. An animal in a low stress condition will alternate between grazing and walking, choosing a direction of walking randomly but biased toward the direction it is pointing. State duration, walking speed and direction are all stochastic. Unstressed animals exhibit very little herding instinct (as observed in the field) until they get very distant from each other. An animal that is experiencing high stress will move toward other animals, and will not resume grazing until its stress has gone down. The stress level of an animal decays over time.

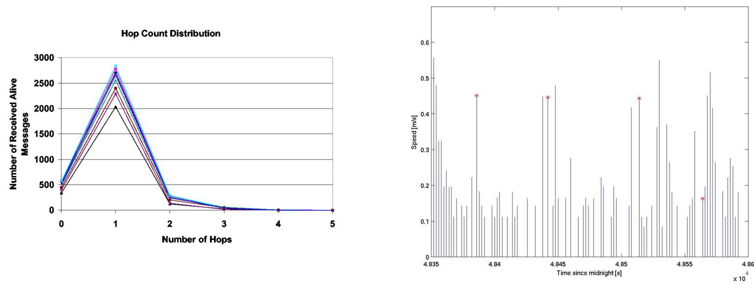
In addition to using stress, the stimulus has an immediate effect on the motion of the animal. We have used two different models, each of which take inspiration from field observations. In the first model, a stimulus causes the animal to quickly turn approximately $90°$. This behavior was also observed in [5]. In the second model, the cow walks forward for a short time when stimulated.

To test the algorithms against these models, we ran virtual fences on a simulated herd with widely varying parameters. The overall goal was to move the virtual fence slowly into the herd and test how quickly the herd moved away from the encroaching fence. This was tested with different values for the grazing speed and walking speed of the cows, the level of herd-attraction and the probability that a stimulus would have the desired effect. We found that the parameters affected the overall speed of the herd in front of the fence and the number of stimuli that were applied, but in all cases the herd did move in the desired direction.

4.2 Cow experiments

We performed a series of field experiments at Cobb Hill, targeting four issues: (1) collecting data to create a grazing model for the cows, which is used in the fence control algorithm; (2) collecting connectivity data and information propagation data, which is used to determine the multi-hop routing method for networking the herd; (3) collecting stimulus-response data for individual animals; and (4) collecting response data for the virtual fence on a group of animals.

Our preliminary results are encouraging. Animals respond to artificial potentials of sounds generated by the virtual fence by moving forward if they are on their own (see Figure 4(b)), or toward the group if they are in close proximity to the group. The animals responded to sounds (see Figure 4(b)) but habituation to stimuli was a problem. Others[2] have combined the sounds with shocks to avoid habituation.

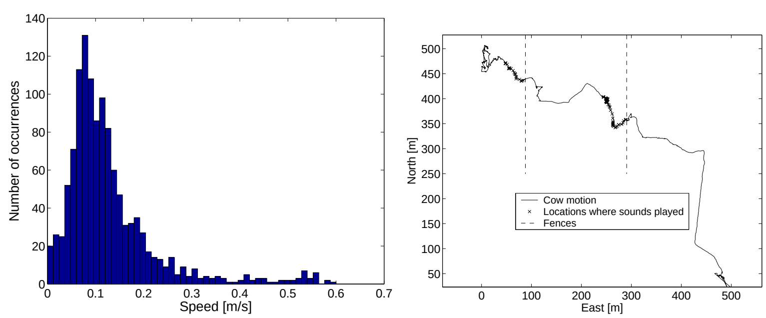


(a) (b)

Fig. 4. Experimental results. (a) Connectivity data (b) Speed when sounds are played.

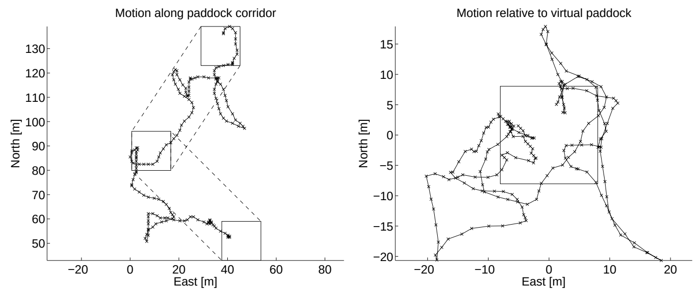
Our later experiments supported the two-state grazing/walking model, as can be seen in a speed histogram of one animal, Fig. 5(a). These data show that the cows have a wide range of speeds throughout the day, although the distribution is not exactly bimodal. Instead we see that they spend a large amount of their time moving quite slowly, and the rest of the time at higher, but differing, speeds.

We also successfully deployed a static virtual fence. As can be seen in Fig. 5(b), the cow was not deterred from passing through the fence, however it did slow down (to a statistically signifi cant degree) while the sounds were playing.



(a) (b)

Fig. 5. More experimental results. (a) Speed histogram (b) Trace past a fence with sounds.



(a) (b)

Fig. 6. Dynamic fence experiment results. (a) Absolute and (b) relative motion of one person.

4.3 Dynamic Fence Experiments

We have tested the dynamic fence algorithm using the smart collars on people[2]. Plans were created in the planning GUI and exported directly to the collars. The results were successful algorithmically, in that the fences appeared in the correct real-world locations at the desired times, but were less successful from a herding standpoint. Motion traces for one person and the group of four people, both absolute and relative to the moving paddock, are shown in Fig. 6. It is interesting to note that the general motion of the people was along the desired corridor, as in Fig. 6(a), despite spending very little time in the virtual paddock, as seen in Fig. 6(b). This may be biased by the experimenters walking in the general direction of the virtual paddock motion.

One benefit of this experiment was that the subjects were able to verbalize their feelings about the system. Primarily, they felt that the resolution of the stimulus gradient was insufficient to be of assistance. Also, they tended to be inquisitive, actively exploring the acceptable boundaries of their space. Together with relatively high walking speeds (averaging over 0.5 m/s) they were able to move a large distance between sounds and were unable to relocate the virtual paddock. We believe that with a more appropriate stimulus (such as Anderson's stimulus for cows[2]) that this can be overcome.

5 Conclusion

Virtual fencing is a concept that could radically change the way that humans manage farm animals. We have developed a herd-animal simulator based on individual state machines and potential fields which exhibit many characteristics seen in real animal groups. Control techniques developed using the simulator have been tested on a

[2] We will extend our experiments to animals as soon as the weather allows them to return to the pastures.

small herd of 10 animals using using off-the-shelf hardware. Our physical experiments show promise for this application but much work remains to be done. The range of sound stimuli we tried were not very effective, although we did observe some reactions. Other researchers [1] have achieved consistent response to sound followed by electric shock stimuli, but this was not an option available to us. There are signifi cant open questions related to stimulus habituation and the need for explicit training which will require close collaboration between roboticists and animal behaviorists.

Acknowledgments

We thank Simon Holmes a Court, Steve Petty, and the staff at Heytesbury Beef Australia for inspiring us to think about this problem and for great technical insights into cows and herding. We especially thank Steven, Kerry and Paul at Cobb Hill for their assistance and enthusiasm and the use of their cows for these experiments. We thank Lorie Loeb for facilitating our work. Tom Temple assembled the cow collars. Jenni Groh provided invaluable advice on the experimental procedure. The experimental part of this work has been done under protocol assurance A3259-01 given by the Institutional Animal Care and Use Committee (IACUC) of Dartmouth College. Zack Butler was supported under Award No. 2000-DC-CX-K001 from the Offi ce for Domestic Preparedness, U.S. Department of Homeland Security. Points of view in this document are those of the authors and do not necessarily represent the offi cial position of the U.S. Department of Homeland Security.

References

1. D. Anderson. Personal communication on may 3, 2004, 2004.
2. D.M Anderson, C.S. Hale, R. Libeau, and B. Nolen. Managing stocking density in real-time. In N. Allsop, A.R. Palmer, S.J. Milton, K.P. Kirkman, G.LH. Kerley, C.R. Hurt, and C.J. Brown, editors, *Proc VII International Rangelands Conf.*, pages 840–843, Durham, South Africa, August 2003.
3. Z. Butler, P. Corke, R. Peterson, and D. Rus. Virtual fences for controlling cows. In *ICRA*, New Orleans, LA, 2004.
4. R.E. Marsh. Fenceless animal control system using gps location information. Technical Report US Patent 5,868,100, Agritech Electronics, February 1999.
5. T.M. Quigley, H.R. Sanderson, A.R. Tiedemann, and M.K. McInnis. Livestock control with electrical and audio stimulation. *Rangelands*, June 1990.
6. S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentics Hall series on Artificial Intelligene, 2002.
7. A.R. Tiedemann, T.M. Quigley, L.D. White, W.S. Lauritzen, J.W. Thomas, and M.K. McInnis. Electronic (fenceless) control of livestock. Technical Report PNW-RP-510, United States Department of Agriculture, Forest Service, January 1999.
8. R. Vaughan, N. Sumpter, A. Frost, and S. Cameron. Robot sheepdog project achieves automatic flock control. *Proc. Fifth International Conference on the Simulation of Adaptive Behaviour.*, 1998.