Chapter 7
Swarm and Entropic Modeling for Landmine Detection Robots

Cagdas Bayram, Hakki Erhan Sevil, and Serhan Ozdemir

7.1 Introduction

Even at the dawn of the 21st century, landmines still pose a global threat. Buried just inches below the surface, combatants and noncombatants alike are all at risk of stepping on a mine. Their very nature is such that these furtive weapons do not discriminate, making it an urgent task to tackle the problem. According to the U.S. State Department [1], based on an estimate reported just a few years ago, there are well over 100 million anti-personnel mines around the world. The existence of these passive weapons causes a disruption in the development of already impoverished regions, as well as maiming or killing countless innocent passers-by. Since the ratification of the anti-personnel mine total ban treaty in 1997, their detection, removal, and elimination have become a top priority. Nevertheless, at the current rate, given the manpower and the man-hours that could be dedicated to the removal of these sleeping arms, it would take centuries. The concerns regarding the speed of removal and safety of the disposers eventually bring us to the discussion of the proposed method.

There are numerous efforts for utilizing robots for landmine detection and/or removal whereas nearly all research activity based on robotics seems to be focused on using sophisticated systems with costly hardware [2]. Even then, the speed that these robots can offer for mine detection is limited because the high associated cost limits the number of robots procured. Another shortcoming of a complex system is the difficulty of repair and maintenance in a harsh environment such as a minefield and also the possible catastrophic loss of the entire system due to an unexpected mine detonation.

As opposed to the idea of having a complete agent with state-of-the-art equipment, the goal may be accomplished by down-to-earth individuals working as a team, indirectly guided by a competent alpha agent. The task of the swarm is to autonomously sweep an area for mines as quickly as possible, as safely as possible. The swarm should be scalable and robust: loss or gain of members should not affect the behavior and reliability of the system, and obstacles or any other disturbances

Oscar Castillo et al. (eds.), *Trends in Intelligent Systems and Computer Engineering.*
© Springer Science+Business Media, LLC 2008

should not affect the stability. Following these guidelines, the main objective of this study is to present an efficient autonomous navigation and detection method to guide a group of inexpensive robotic agents. To lower the cost of the agent, a minimal number of sensors, actuators, Ics, and other components should be used. In addition, the navigation method should fare well without needing very precise (and costly) sensors.

Nature already provided good solutions to manage groups of less able beings: fish schools, ant swarms, animal packs, bird flocks, and so on. With the growing desire of humans to create intelligent systems, these biosystems are being thoroughly inspected [3–10] and implemented [11–14] in various studies.

In this study a robotic agent is referred to as a drone, the group of robotic agents is referred to as a swarm, and the agent with mapping abilities is referred to as the alpha drone.

7.2 Desired Swarm Behavior and Drone Properties

Before going into the details of anything else, one should define the desired behavior of the swarm. The swarm should:

- Autonomously sweep a prescribed area.
- Exhibit swarming (collision avoidance, polarization, attraction to swarm mates).
- Designate the mine locations with an acceptable accuracy.
- Find all the mines in the swept area (high reliability).
- Be able to tolerate loss of members due unexpected situations.

At this point, equipping all members with advanced sensors and microcomputers will quite increase the cost, so it is decided to have two types of agents: drones and an alpha drone. Because our main interest is to have minimalist robotic agents that could be fielded in large numbers to speed up the mine cleaning process, a drone should:

- Have a unique identification number.
- Know and control its heading and speed.
- Have a means of wide-angle proximity detection (i.e., sonar array). These sensors need not be very precise. The behavior model should work for rough and noisy sensor readings.
- Have a means of detecting mines (i.e., metal detector).
- Have a means of wireless communication although it should consume low power, be inexpensive, and therefore low-range.
- Have the means of making simple preprogrammed decisions.
- Avoid stepping on mines.

The alpha drone's main task is to record the mine locations and indirectly control the drones by presenting them with a desired heading; the alpha drone should:

- Know its absolute location with good accuracy (i.e., using GPS).

- Know the boundaries of the area to be swept.
- Be able to indirectly force the drones to move in a direction.
- Collect landmine location data from drones and mark them on a map.
- Never step on a mine.
- Have all the necessary subsystems of a drone.

7.3 Drone Model

There are some proposed distributed behavior models for fish schools and bird flocks. Our particular interest is in the models proposed by Aoki [8], Huth and Wissel [9], Couzin et al. [10], and Reynolds [11]. To summarize, schooling and flocking was explained using three concentric zones: a zone of repulsion, zone of orientation, and zone of attraction. Also it was shown that the overall heading of the flock can be controlled by adding a migratory urge, which is simply a direction. The models not only explain the schooling phenomena to a good extent but also give a good tool to manage groups of robots. In a previous simulation work by authors based on these fish school models, it has been seen that the school tends to move in a hexagonal close-packed formation. This is an ideal formation pattern to be used in mine sweeping because there are no gaps left in a group of mine detectors. The authors began with these preliminary models, altered them to fit the world of mobile robots by translating the means of sensing and locomotion, and extended the model further.

Perhaps the most important problem of adapting these originator models to the world of robotics is the means of sensing. In the biological world, thanks to millions of years of evolution, even the simplest organism is equipped with highly precise and effective sensors. However, the robotic systems still have to utilize relatively poor sensors compared to those of biological organisms. Despite the advances in image processing and pattern recognition techniques, a full-blown visual sensor is still too costly or merely incompetent to deal with the complicated real world. The proposed model in this study is especially devised for robotic agents equipped with simple, readily available, and well-understood sensors such as infrared transceivers.

In this study we call an individual mobile robotic agent a drone. A drone is a simple entity, trying to find its way following the alpha drone's migration orders and mimicking other drones' movement while trying to survive. In our basic model, a drone has an array of near-range proximity sensors (possibly ultrasonic), a low-range wireless transmitter/receiver (possibly RF), differential locomotion (i.e., tracks), a simple microcontroller, a mine detector, a digital compass, and an attraction beacon (possibly an IR beacon with a certain frequency).

The drone model implemented in simulation is given in Fig. 7.1. Radii r_{PSA}, r_{MDS}, r_{AB}, and r_{WCS} are ranges for the proximity sensor array (PSA), mine detection sensor (MDS), attraction beacon (AB), and wireless communication system (WCS), respectively, where l_{MDS} represents the distance where the mine detection sensor is placed apart from the robot body. The hatched circle represents the robot body.

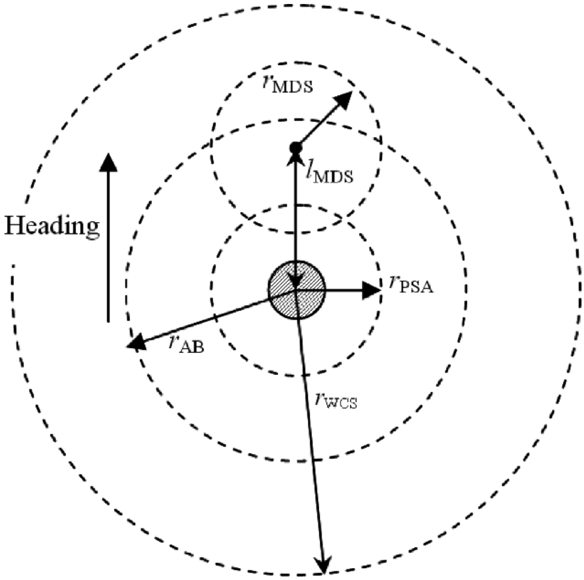

Fig. 7.1 Proposed drone model

The PSA is an active sensor array that gives true/false outputs within a certain angle resolution; a drone has a rough idea of the bearings of nearby objects. MDS gives an analog reading; in the case of a metal detector the output will be higher when a metallic object is closer and vice versa. WCS has two bidirectional channels, one being used for communication with the alpha drone (alpha channel) and the other with drones (beta channel). A drone also knows and can change its speed and heading. The attraction sensor array (ASA) is a passive sensor that detects other AB signals within a certain angle resolution; a drone has a rough idea of the bearings of other drones.

WCS broadcasts the following information from the beta channel in specific time intervals: a unique ID number, its speed, and heading. AB emits a unidirectional, "I am here," signal at specific time intervals. When MDS detects a mine, WCS broadcasts its ID along with a, "Mine detected," message and a life count from the alpha channel. All drones rebroadcast any message they receive from the alpha channel, coming both from drones and the alpha drone, after decreasing the life count by one. A message with zero life count is not broadcast.

This system is a simplified version of the packet routing method used in the Internet protocol and eliminates the possibility of unendingly broadcasting the same message over and over. It's important to understand that it may be difficult and impractical to precisely synchronize the "clocks" of the drones, hence the so-called WCS broadcasts will occur asynchronously. Another interesting point is that beta channel broadcasts may also be forwarded like alpha channel broadcasts, thus enabling a drone to know all members' current velocity. However this will result in much more crowded network traffic and may not be applicable in practice for large swarms. In simulation, both cases are considered.

7.4 Alpha Drone Model

The alpha drone is nothing more than a drone with two additional subsystems as a GPS and a means of knowing the relative position of drones in the flock. One such method is proposed by Wildermuth and Schneider [14] based on vision and pattern recognition. Also, if the wireless communication system is selected to operate on RF, triangulation techniques may be used to obtain the relative position data. The alpha drone has two main tasks: to present a general heading, the migratory urge, for the swarm, and to mark the mine locations on a map that is reported by drones and detected by the alpha drone itself. In addition, the alpha drone exhibits swarming as do other drones. Ultimately, the alpha drone requires more computational power and memory.

In case detecting the relative position of drones becomes too complicated or too slow, certain other approaches may be used. (1) Whenever a mine is detected, the alpha drone marks the place where it is currently located. The mine map generated will only give a density distribution of the minefield, without giving the actual coordinates. (2) All drones are equipped with GPS, which may increase the cost to undesirable levels.

7.5 Distributed Behavioral Rules and Algorithm

The behavior of drones can be divided into two categories: migrating and swarming, and mine detection and avoidance (see Fig. 7.2). These behavior modes are fused by a decision-making process. All inputs from subsystems are multiplied by weights and a resultant velocity request is generated. Finally, the velocity request is fed into the traction system to generate motion. The inputs are composed of PSA, MDS, and ASA readings, the heading imposed by the alpha drone, and an average of received velocity broadcasts by other drones.

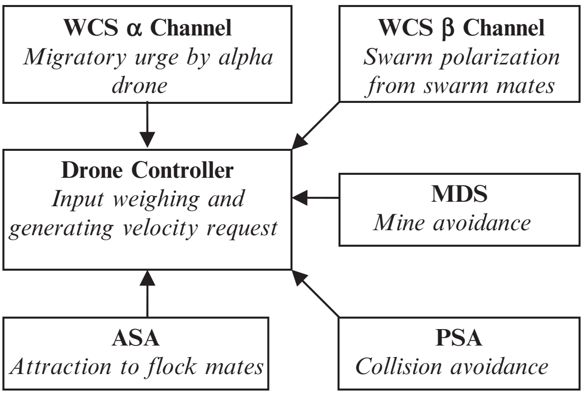

Fig. 7.2 Drone subsystems

Assume that a drone is able to fully perceive its surroundings, thus knowing the exact locations of obstacles. To exhibit basic collision avoidance, the drone should move in the opposite direction to the sum of unit position vectors (in its local coordinate frame) of obstacles.

In Fig. 7.3, O is the local coordinate frame for a drone; U_1 and U_2 are the unit vectors pointing towards obstacles. For the general case:

$$\hat{\mathbf{u}}_c = -\frac{\sum_{i=1}^{n} \mathbf{u}_i}{\left|\sum_{i=1}^{n} \mathbf{u}_i\right|} \tag{7.1}$$

where n is the number of obstacles, $\mathbf{u}_i$ is the unit vector pointing towards the ith obstacle, and $\hat{\mathbf{u}}_c$ is the unit vector pointing towards the required direction of motion to avoid collision.

In our model, a drone has a specific number of proximity sensors n_{PSA} that are placed symmetrically on a circle. Each sensor is assumed to cover an angle equal to $2\pi/n_{PSA}$. In addition, these sensors are not able to detect the distance to an obstacle but just provide an on/off signal if something is detected or not in a certain range.

Figure 7.4 shows a PSA with $n_{PSA} = 6$. For this particular example, each sensor covers an area of $2\pi/6 = 60°$. If an obstacle comes in PSA range within 0–59° then the first sensor is activated, within 60–129° the second sensor is activated, and so on. Because a sensor does not indicate the exact bearing of the obstacle, it is assumed that the obstacle is just in the middle of the sensor coverage. That is, if the first

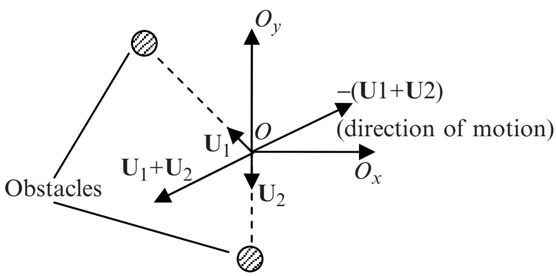

Fig. 7.3 Basic collision avoidance

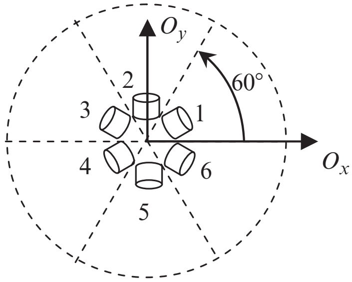

Fig. 7.4 PSA with six sensors

sensor is activated, we assume that the obstacle is at 30°, for the second sensor it is 90°, for the ith sensor it is

$$\phi_{obstacle} = \frac{2\pi}{n_{PSA}}(i-0.5)$$

Because we are dealing only with unit vectors in Eq. 7.1, it's enough to find the polar angle of $\hat{\mathbf{u}}_c$. This angle is our heading to avoid collision that is given by

$$\begin{aligned}\theta_c &= -\text{ATAN2}\left[\sum_{i=1}^{n_{PSA}} \sin(q_i)\,,\ \sum_{i=1}^{n_{PSA}} \cos(q_i)\right]\\ q_i &= \frac{2\pi\mu_i(i-0.5)}{n_{PSA}}\end{aligned} \tag{7.2}$$

where μ_i is the respective sensor output as 1 or 0 (1 if the respective sensor detects an object, 0 otherwise), and θ_c is the collision avoidance heading request.

The attraction heading request is derived exactly as collision avoidance with a single exclusion. The minus sign is removed because we want the drone to move towards the other drones. Also note that ASA is a passive sensor and it only detects the signal emitted by other drones.

$$\begin{aligned}\theta_a &= -\text{ATAN2}\left[\sum_{i=1}^{n_{ASA}} \sin(q_i),\ \sum_{i=1}^{n_{ASA}} \cos(q_i)\right]\\ q_i &= \frac{2\pi\mu_i(i-0.5)}{n_{ASA}}\end{aligned} \tag{7.3}$$

where n_{ASA} is the number of sensors in attraction sensor array, μ_i is the respective sensor output as 1 or 0, and θ_a is the attraction heading request.

The heading request for migration is supplied by the alpha drone; because a drone knows its heading relative to true north, this migration direction is simply converted to a heading request.

The swarm polarization heading is generated by summing the broadcasts from other drones. The main problem is that the broadcasts are asynchronous. We have two solutions to this problem: use a fixed length array in memory to keep the incoming broadcasts; or use a dynamic array (stack) with a specified maximum size, add each incoming broadcast to the stack with a timestamp, and delete broadcasts that are older than a certain time. The second approach is used in simulation. The broadcasts of drones are in the same manner as the migration urge broadcast of the alpha drone, but in this case drones broadcast their actual heading in terms of compass directions.

$$\theta_p = \sum_{i=1}^{n} \frac{\theta_i}{n} \tag{7.4}$$

where n is the number of elements in polarization stack, and θ_i is the heading data in the ith stack element.

$$\begin{aligned}\theta_r = \text{ATAN2}(&\eta_m \text{S}\theta_m + \eta_p \text{S}\theta_p + \eta_a \text{S}\theta_a + \eta_c \text{S}\theta_c - \eta_l \mu_l,\\ &\eta_m \text{C}\theta_m + \eta_p \text{C}\theta_p + \eta_a \text{C}\theta_a + \eta_c \text{C}\theta_c)\end{aligned} \tag{7.5}$$

where C and S stands for cosine and sine; $\eta_m, \eta_p, \eta_a, \eta_c, \eta_l$ are the weights of importance for migration, polarization and attraction, collision avoidance, and mine avoidance, respectively; and $\theta_m, \theta_p, \theta_a, \theta_c$ are the heading requests generated by said behaviors. μ_l is the signal strength of MDS. Equation 7.5 is in fact scaling and addition of unit vectors describing behaviors. Another point is that, for example, by selecting η_l and η_c much bigger then the others, the system behavior shifts to hierarchical where survival supersedes all other rules. Only in the absence of mines or obstacles, do the other factors come into effect.

Now that the drone knows where to turn, it needs to know how fast it should go. The guidelines for speed selection can be given as: (1) the fewer drones you see around, go faster to catch up with the flock, and (2) try to move with the same speed as the other flock mates, which helps polarization. At this point, the same type of stack, which is used to store bearing broadcasts, is used to store velocity broadcasts.

$$v' = f(m)\left(\lambda \sum_{i=1}^{n} \frac{v_i}{n} + (1-\lambda)v_{\max}\right) \quad f(m) = \left| \begin{array}{l} m/a,\ m > a \\ 1,\ m \le a \end{array} \right. \tag{7.6}$$

where $f(m)$ is pseudo-acceleration, n is the number of elements in the stack, v_i is the speed data in ith stack element, λ is the polarization parameter, $v_{\max}$ is the maximum attainable speed, m is the number of inactive sensors in ASA (i.e., sensors not detecting anything), and a is a limitation value that prevents too much speed loss for members near the center of the flock. Note that $1 \le a <$ number of sensors in ASA and $a = 1$ means no speed loss limitation. Also note that $0 \le \lambda < 1$, for $\lambda = 0$ speed matching will not occur.

7.6 Simulation Results

Simulations were carried out in two phases. In the first, interactions between two individuals are taken into consideration. The idea was whether a concept of swarm stability could be specified. Among any collection of individuals, the quality of being a swarm is inversely proportional to the distance between the particles, or agents. There is such a distance that the agents are not in a swarm formation any more but rather are acting freely. This first phase defines a "swarm stability" or swarm entropy, which is quantified in Tsallis entropy, whose details may be explained in a separate study. Agents normally roam apart from each other in search of food (a mine) so as to cover an area as fast as possible with the least likelihood of missing anything during the search. But this separation should not be too great in order not to lose the swarm behavior along with all the advantages that accompany it.

The definition of entropy, which was first discovered by Ludwig Boltzmann, can be given as a measure of disorder. There are many types of entropy definition in the literature. One of them is Tsallis entropy, first explained in 1988 [15]. Tsallis modified the mathematical expression of entropy definition in his study and defined a new parameter, q.

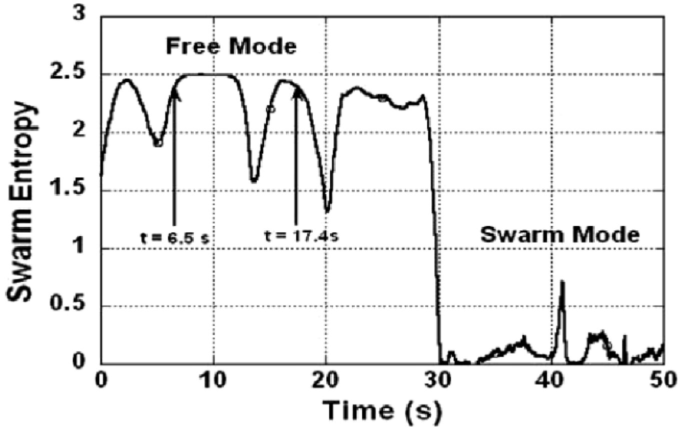

Fig. 7.5 Entropy changing with time for free and swarm modes

$$S_T = \frac{1 - \sum_i P_i^q}{1 - q} \tag{7.7}$$

During the roaming of the individuals, entropy fluctuates at around its highest, meaning the distance between the agents is rather far, and swarm stability is low; that is, the quality of remaining as a swarm may disappear should the particles get farther away. However, a sudden decrease in entropy may occur as in Fig. 7.5, when the individuals converge after the discovery of food (a mine).

Peaks in the entropy of a swarm mode were created by attractive and repulsive behavior of individuals. Close encounters are considered as risks of collision and quick reactions to avert it. Please note that such peaks are missing in the free mode (i.e., roaming a certain field).

In Fig. 7.6, at around 62.2 s, a repulsion may be seen because these two agents have moved too close, namely into the repulsive field, and at 62.7, they start to move back again.

In the second phase, the whole model is implemented in a computer program in an object-oriented fashion. Lengths are described in terms of "units." The following drone parameters are used for each simulation run (Fig. 7.7).

Maximum speed: 20 units/s	**Turning rate: 180 °s**
Drone shape: disc	Diameter: 10 units

Four distinct swarm behaviors are observed. These are: high polarization (HP), balanced polarization (BP), low polarization (LP), and disarray. The first three behaviors have their uses where the disarray behavior indicates an unstable swarm, which is not desired.

High polarization means that the velocity (both speed and heading) of an individual drone is nearly the same as the swarm average in the absence of disturbances. The average speed of the swarm is maximized. The main disadvantage is that the swarm aggregates very slowly when it meets a disturbance (an obstacle or a mine).

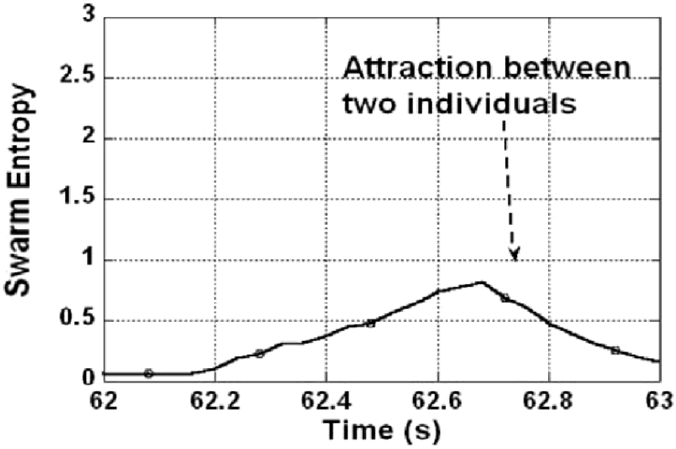

Fig. 7.6 Entropy variation where there is an attraction between two individuals

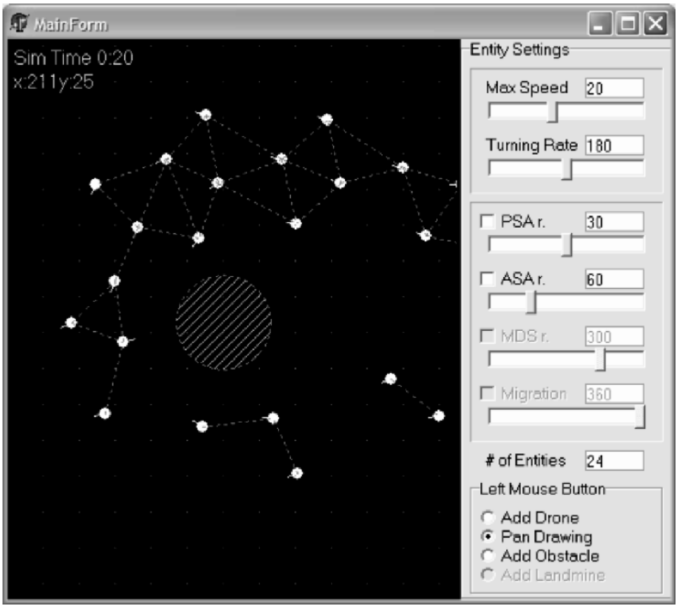

Fig. 7.7 Simulation screenshot: drones detected a mine. The "Landmine Detection Simulator" can be found at the author's Web site: http://www.iyte.edu.tr/∼erhansevil/landmine.htm

Thus the mine detection reliability is decreased significantly. This behavior results in either a high migration weight μ_m, or a high polarization weight μ_p. It is an ideal swarm behavior for traversing mine-safe zones to go quickly to an objective area.

Balanced polarization means that the velocity of an individual drone is close to the neighboring drones but not necessarily close to the average swarm velocity. This offers high speed (although lower then HP) and high reliability. The swarm aggregates quickly after meeting disturbances. This behavior results in nearly equal μ_m and μ_p, and high μ_a. It is an ideal swarm behavior for most cases.

Low polarization means that the velocity of an individual drone is highly different from that of its neighbors. This happens when μ_a is high and μ_p is low and also μ_m is selected between. The only use for this behavior is that the swarm can find its way when there are too many obstacles, such as a labyrinth.

Disarray occurs if:

- μ_a is too low (swarm disintegrates).
- μ_l is too low (drones step on mines).
- μ_c is too low (drones collide with each other).
- μ_m is too low (swarm moves in a random direction).

Note that, by really unsuitable parameters, more than one symptom of disarray can be observed. Surprisingly, if the other parameters are chosen well, a low μ_p, even zero, does not lead to disarray.

Another important concept is efficiency. What should be the optimal number of drones to be used? It is observed that up to an optimum population, efficiency of the swarm increases. After that point, adding more drones does not improve the mine detection speed or performance much. This is mainly because too many drones form a useless bulk in the center of the swarm. However, the optimum number of drones also depends on the terrain (rough, smooth, etc.), landmine density, actual speed and turning rates of drones, sensor ranges, and swarm behavior.

7.7 Conclusion

A distributed behavioral model to guide a group of minimalist mobile robots is presented. The main point of interest for the model is that it is based on weighting sensor inputs and not on precedence-based rules. By chancing the weights, it is possible to shift the behavior of the swarm while all other physical parameters (such as sensor ranges) remain constant. The model is presented in a computer simulation that gave promising results.

It should be noted that the selection of weights changes the behavior of the swarm drastically and sometimes unexpectedly. To optimize the drone behavioral weights is the upcoming part of this study on which the authors are currently working.

References

1. US State Department (1998) *Hidden Killers*: The Global Landmine Crisis. Report released by the U.S. Department of State, Bureau of Political-Military Affairs, Office of Humanitarian Demining Programs, Washington, DC, September 1998.
2. Huang QJ, Nonami K (2003) Humanitarian mine detecting six-legged walking robot and hybrid neuro walking control with position/force control. *Mechatronics* 13: 773–790.

3. Dolby AS, Grubb TC (1998) Benefits to satellite members in mixed-species for aging groups: An experimental analysis. *Animal Behavior* 56: 501–509.
4. Adioui M, Treuil JP, Arino O (2003) Alignment in a fish school: A mixed Lagrangian–Eulerian approach. *Ecological Modelling* 167: 19–32.
5. Cale PG (2003) The influence of social behaviour, dispersal and landscape fragmentation on population structure in a sedentary bird. *Biological Conservation* 109: 237–248.
6. Smith VA, King AP, West MJ (2002) The context of social learning: Association patterns in a captive flock of brown-headed cowbirds. *Animal Behavior* 63: 23–35.
7. Green M, Alerstam T (2002) The problem of estimating wind drift in migrating birds. *Theoretical Biology* 218: 485–496.
8. Aoki I (1982) A simulation study on the schooling mechanism in fish. *Social Science Fish* 48: 1081–1088.
9. Huth A, Wissel C (1992) The simulation of the movement of fish schools. *Theoretical Biology* 156: 365–385.
10. Couzin D, Krause J, James R, Ruxton GD, Franks NR (2002) Collective memory and spatial sorting in animal groups. *Theoretical Biology* 218: 1–11.
11. Reynolds, Flocks CW (1987) Herds and schools: A distributed behavioral model. *Computational Graph* 21: 25–34.
12. Sugawara K, Sano M (1997) Cooperative acceleration of task performance: Foraging behavior of interacting multi-robots system. *Physica D* 100: 343–354.
13. Martin M, Chopard B, Albuquerque P (2002) Formation of an ant cemetery: Swarm intelligence or statistical accident? *FGCS* 18: 951–959.
14. Wildermuth D, Schneider FE (2003) Maintaining a common co-ordinate system for a group of robots based on vision. *Robotics and Autonomous Systems* 44: 209–217.
15. Tsallis C (1988) Possible generalization of Boltzmann–Gibbs statistics. *Journal of Statistical Physics* 52: 479–487.