An Overview of Physicomimetics

William M. Spears, Diana F. Spears,
Rodney Heil, Wesley Kerr, and Suranga Hettiarachchi

Computer Science Department,
University of Wyoming, Laramie, WY, 82070, USA
wspears@cs.uwyo.edu
WWW home page: http://www.cs.uwyo.edu/∼wspears

Abstract. This paper provides an overview of our framework, called *physicomimetics*, for the distributed control of swarms of robots. We focus on robotic behaviors that are similar to those shown by solids, liquids, and gases. Solid formations are useful for distributed sensing tasks, while liquids are for obstacle avoidance tasks. Gases are handy for coverage tasks, such as surveillance and sweeping. Theoretical analyses are provided that allow us to reliably control these behaviors. Finally, our implementation on seven robots is summarized.

1 Vision

The focus of our research is to design and build rapidly deployable, scalable, adaptive, cost-effective, and robust swarms of autonomous distributed robots. Our objective is to provide a scientific, yet practical, approach to the design and analysis of swarm systems.

The team robots could vary widely in type, as well as size, e.g., from nanobots to micro-air vehicles (MAVs) and micro-satellites. A robot's sensors perceive the world, including other robots, and a robot's effectors make changes to that robot and/or the world, including other robots. It is assumed that robots can only sense and affect nearby robots; thus, a key challenge has been to design "local" control rules. Not only do we want the desired global behavior to emerge from the local interaction between robots (self-organization), but we also require fault-tolerance, that is, the global behavior degrades very gradually if individual robots are damaged. Self-repair is also desirable, in the event of damage. Self-organization, fault-tolerance, and self-repair are precisely those principles exhibited by natural physical systems. Thus, many answers to the problems of distributed control can be found in the natural laws of physics.

This paper provides an overview of our framework for distributed control, called "physicomimetics" or "artificial physics" (AP). We use the term "artificial" (or virtual) because although we are motivated by natural physical forces, we are not restricted to them [1]. Although the forces are virtual, robots *act* as if they were real. Thus the robot's sensors must see enough to allow it to compute the force to which it is reacting. The robot's effectors must allow it to respond to this perceived force.

There are two potential advantages to this approach. First, in the real physical world, collections of small entities yield surprisingly complex behavior from very simple interactions between the entities. Thus there is a precedent for believing that complex control is achievable through simple local interactions. This is required for very small robots, since their sensors and effectors will necessarily be primitive. Second, since the approach is largely independent of the size and number of robots, the results scale well to larger robots and larger sets of robots.

2 The Physicomimetics Framework

The basic AP framework is elegantly simple. Virtual physics forces drive a multi-robot system to a desired configuration or state. The desired configuration (state) is one that minimizes overall system potential energy. In essence the system acts as a molecular dynamics ($\boldsymbol{F} = m\boldsymbol{a}$) simulation.

At an abstract level, AP treats robots as physical particles. This enables the framework to be embodied in robots ranging in size from nanobots to satellites. Particles exist in two or three dimensions and are point-masses. Each particle i has position $\boldsymbol{x}$ and velocity $\boldsymbol{v}$. We use a discrete-time approximation to the continuous behavior of the system, with time-step Δt. At each time step, the position of each particle undergoes a perturbation $\Delta\boldsymbol{x}$. The perturbation depends on the current velocity, i.e., $\Delta\boldsymbol{x} = \boldsymbol{v}\Delta t$. The velocity of each particle at each time step also changes by $\Delta\boldsymbol{v}$. The change in velocity is controlled by the force on the particle, i.e., $\Delta\boldsymbol{v} = \boldsymbol{F}\Delta t/m$, where m is the mass of that particle and $\boldsymbol{F}$ is the force on that particle. A frictional force is included, for self-stabilization. This is modeled as a *viscous friction* term, i.e., the product of a viscosity coefficient and the robot's velocity (independently modeled in the same fashion by Howard et al. [2]). We have also included a parameter F_{max}, which restricts the maximum force felt by a particle. This provides a necessary restriction on the acceleration a robot can achieve. Also, a parameter V_{max} restricts the velocity of the particles, which is very important for modeling real robots.

Given a set of initial conditions and some desired global behavior, it is necessary to define what sensors, effectors, and local force laws are required for the desired behavior to emerge. This is explored, in the next section, for a variety of simulated static and dynamic multi-robot configurations. Our implementation with robots is discussed in Section 3.2.

3 Physicomimetic Results

Our research has focused on robotic behaviors that are similar to those shown by solids, liquids, and gases. Solid crystalline formations are useful for distributed sensing tasks, to create a virtual antenna or synthetic aperture radar. For such tasks it is important to maintain connectivity and a lattice geometry. Liquids are for obstacle avoidance tasks, since fluids easily maneuver around obstacles while retaining connectivity. Solid and liquid behaviors are formed using a similar force law, that has attractive and repulsive components. The transition between solids and liquids can be performed via a change in only one parameter, which balances the attractive and repulsive components [3].

Finally, gases are handy for coverage tasks, such as surveillance and sweeping maneuvers. For these tasks it is imperative that coverage can be maintained, even in the face of individual robot failures. Gas-like behaviors are created using purely repulsive forces.

3.1 Simulation Results

Solids: Our initial application required that a swarm of MAVs self-organize into a hexagonal lattice, creating a distributed sensing grid with spacing R between MAVs [4]. Potential applications include sensing grids for the mapping or tracing of chemical/biological plumes [5] or the creation of virtual antennas to improve the resolution of radar images [1]. To map this into a force law, each robot repels other robots that are closer than R, while attracting robots that are further than R in distance. Thus each robot has a circular "potential well" around itself at radius R – and neighboring robots will be separated by distance R. The intersection of these potential wells is a form of constructive interference that creates "nodes" of low potential energy where the robots are likely to reside. A simple compass construction illustrates that this intersection of circles of radius R will form a hexagonal lattice where the robot separation is R. Note that potential energy (PE) is never actually computed by the robots. Robots compute local force vectors. PE is only computed for visualization or mathematical analysis.

With this in mind, we defined a force law $F = Gm_im_j/r^p$, where $F \leq F_{max}$ is the magnitude of the force between two particles i and j, and r is the distance between the two particles. The variable p is a user-defined power, which ranges from -5.0 to 5.0. Unless stated otherwise, we assume $p = 2.0$ and $F_{max} = 1$ in this paper. Also, $m_i = 1.0$ for all particles (although the framework does not require this). The "gravitational constant" G is set at initialization. The force is repulsive if $r < R$ and attractive if $r > R$. Each particle has one sensor that can detect the distance and bearing to nearby particles. The one effector enables movement with velocity $v \leq V_{max}$. To ensure that the force laws are local, we allow particles to sense only their nearest neighbors. Hence, particles have a visual range of only $1.5R$.

A simple generalization of this force law will also create square lattices. If robots are arbitrarily labeled with one of two colors, then square lattices are formed if robots that have unlike colors have a separation of R, while robots that have like colors have separation $\sqrt{2}R$. Furthermore, transformations between square and hexagonal lattices (and vice versa) are easily accomplished. Figure 1 illustrates formations with 50 robots. The initial deployment configuration (left) is assumed to be a tight cluster of robots. The robots move outwards into a square formation (middle). Then they transform to a hexagonal formation (right). Self-repair in the face of agent failure is also straightforward [6].

The total PE of the initial deployment configuration is an excellent indicator of the quality of the final formation. High PE predicts high quality formations.

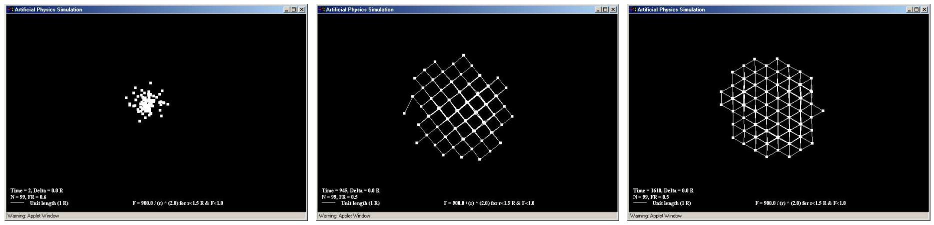

Fig. 1. The initial deployment configuration (left) is assumed to be a fairly tight cluster of robots. The robots move outwards into a square formation (middle). Then they transform to a hexagonal formation (right).

This energy is dependent on the value of G, and it can be proven that the optimal value of G for hexagonal lattices is [7]:

$$G_{opt}{}^{\triangle} = F_{max} R^p [2 - 1.5^{1-p}]^{p/(1-p)} \tag{1}$$

The value of G_{opt} does not depend on the number of particles, which is a nice result. However, for square lattices:

$$G_{opt}{}^{\square} = F_{max} R^p \left[\frac{\sqrt{2}(N-1)[2 - 1.3^{1-p}] + N[2 - 1.7^{1-p}]}{\sqrt{2}(N-1) + N} \right]^{p/(1-p)} \tag{2}$$

Note that in this case G_{opt} depends on the number of particles N. It occurs because there are two classes (colors) of robots. However, the dependency on N is not large and goes to zero as N increases.

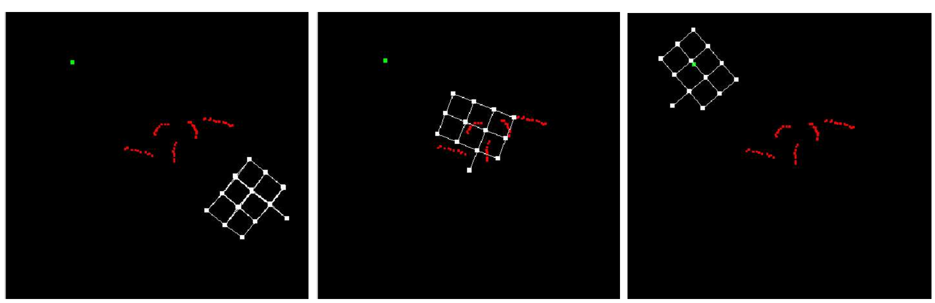

Fig. 2. A solid formation moves through an obstacle field towards a goal (upper left part of the field). The rotations and counter-rotations of the whole collective are an emergent property.

Our current research is focused on the movement of formations through obstacle fields towards some goal. Larger obstacles are created from multiple, point-sized obstacles; this enables flexible creation of obstacles of arbitrary size and shape. As a generalization to our standard paradigm, goals are attractive, whereas obstacles are repulsive (similar to potential field approaches, e.g., [8]).

Figure 2 illustrates how a square formation moves through an obstacle field via a sequence of rotations and counter-rotations of the whole collective. This behavior emerges from the interaction of forces and is not a programmed response. If this cannot be accomplished, the formation may not be able to make further progress towards the goal.

Liquids: As stated above, the difference in behavior between solid formations and liquid formations depends on the balance between the attractive and repulsive components of the forces. In fact, the parameter G once again plays a crucial role. Below a certain value of $G \equiv G_t$, liquid behavior occurs. Above that value, solid behavior occurs. The switch between the two behaviors acts very much like a phase transition. Using a standard *balance of forces* argument we can show that the phase transition for hexagonal lattices occurs at [3]:

$$G_t{}^{\triangle} = \frac{F_{max} R^p}{2\sqrt{3}} \tag{3}$$

The phase transition law for square lattices is:

$$G_t{}^{\square} = \frac{F_{max} R^p}{2\sqrt{2}+2} \tag{4}$$

Neither law depends on the number of robots N, and the difference in the denominators reflects the difference in hexagonal and square geometries. There are several uses for these equations. Not only can we predict the value of G_t at which the phase transition will occur, but we can also use G_t to help design our system. For example, a value of $G \approx 0.9G_t$ yields the best liquid formation, while a value of $G \approx 1.8G_t \approx G_{opt}$ yields the best solid formations.

As mentioned before, liquids are especially interesting for their ability to flow through obstacle fields, while retaining their connectivity. Figure 3 illustrates how a "square" liquid formation moves through the same obstacle field as before. In comparison with the solid formation shown above, far more deformation occurs as the liquid moves through the obstacles. However, the movement is quicker, because the liquid does not have to maintain the rigid geometry of the solid. Despite this, connectivity is maintained. One can easily imagine a situation where a formation lowers G to move around obstacles, and then raises G to "re-solidify" the formations after the obstacles have been avoided.

Gases: The primary motivation for gas behavior is regional coverage, e.g., for surveillance and sweeping. For stealth it is important for individual robots to have an element of randomness, while the emergent behavior of the collective is

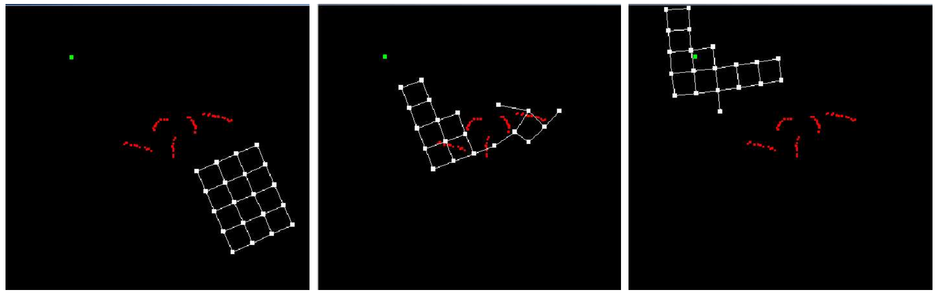

Fig. 3. A liquid formation moves through the same obstacle field towards the goal. Far more deformation occurs, but connectivity is maintained.

still predictable. Furthermore, any approach must be robust in the face of robot failures or the addition of new robots. The AP algorithm for surveillance is simple and elegant – agents repel each other, and are also repelled by perimeter and obstacle boundaries, providing uniform coverage of the region. If robots are added/destroyed, they still search the enclosed area, but with more/less virtual "pressure" [6]. An interesting phase transition for this system depends on the value of G. When G is high, particles fill the corridor uniformly, providing excellent on-the-spot coverage. When G is low, particles move toward the corners of the corridor, providing excellent line-of-sight coverage. Depending on whether the physical robots are better at motion or sensing, the G parameter can be tuned appropriately.

Currently we are investigating the more difficult task of "sweeping" a region, while avoiding obstacles. This task consists of starting a swarm of robots at one end of a corridor-like region, and allowing them to travel to the opposite end, providing maximum coverage of the region in minimal time. A goal force causes the robots to traverse the corridor length. As they move, robots must not only avoid obstacles, but they must also sweep in behind the obstacles to minimize holes in the coverage. One obvious tradeoff is the speed at which the robots move down the corridor. If they move quickly, they traverse the corridor in minimal time, but may move too quickly to sweep in behind obstacles. On the other hand, excellent sweeping ability behind obstacles can significantly slow the swarm. What is required is a Pareto optimal solution that balances sweeping ability with traversal speed $v_{traversal}$.

To address this task we modified our standard AP algorithm to employ a more realistic gas model that has Brownian motion and expansion properties [9]. The collective swarm behavior appears as Brownian motion on a small scale, and as a directed bulk movement of the swarm when viewed from a macroscopic perspective. The expansion properties provide across-corridor coverage and the ability to sweep in behind obstacles. An analogy would be the release of a gas from the ceiling of the room that has an atomic weight slightly higher than the normal atmosphere. This gas drifts downward, moving around obstacles, and expanding back to cover the areas under the obstacles.

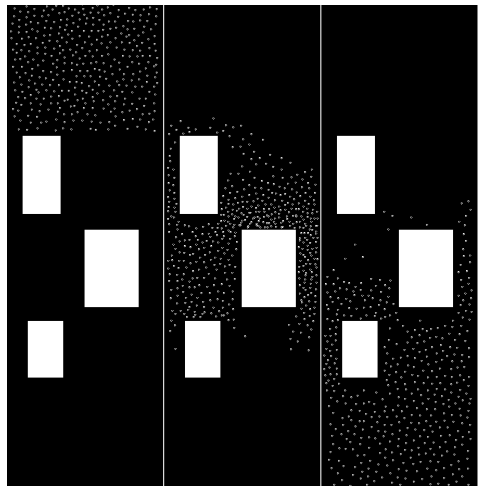

Fig. 4. These three figures depict a sweep of a swarm of robots from the top of a corridor to the bottom.

As mentioned above, speed of movement down the length of the corridor is governed by $v_{traversal}$. However, the expansion properties (across the corridor width) are governed by a temperature parameter T, which determines the *expected* kinetic theory speed [9]:

$$\langle v_{kt} \rangle = \frac{1}{4}\sqrt{\frac{8\pi kT}{m}} \tag{5}$$

where k is Boltzmann's constant. Note that $\langle v_{kt} \rangle$ is an emergent property of the system – each robot can continually change its velocity, based on "virtual" robot/robot, robot/obstacle, and robot/corridor collisions. The net effect is to provide a stochastic component to each robot, while maintaining predictable collective behavior. The resultant velocity of each robot depends on both $v_{traversal}$ and $\langle v_{kt} \rangle$. In other words, although the speed of the swarm is predictable, the individual robot velocities are *not*. This is especially valuable for stealthy surveillance.

Figure 4 illustrates the compromise between traversal speed and the quality of the sweep, providing effective coverage in reasonable time, with the exception of small gaps behind the obstacles. Numerous experiments with different corridors confirm this effectiveness in simulation [10].

3.2 Results with Robots

The current focus of this project is the physical embodiment of AP on a team of robots.

For our experiments, we built seven robots. The "head" of each robot is a sensor platform used to detect other robots in the vicinity. For distance information we use Sharp GP2D12 IR sensors. The head is mounted horizontally on a servo motor. With 180° of servo motion, and two Sharp sensors mounted on opposite sides, the head provides a simple "vision" system with a 360° view. After a 360° scan, object detection is performed. A first derivative filter detects object boundaries, even under conditions of partial occlusion. Width filters are used to ignore narrow and wide objects. This algorithm detects nearby robots, producing a "robot" list that gives the bearing/distance of neighboring robots.

Once sensing and object detection are complete, the AP algorithm computes the virtual force felt by that robot. In response, the robot turns and moves to some position. This "cycle" of sensing, computation and motion continues until we shut down the robots or they run out of power. Figure 5 shows the AP code. It takes a robot neighbor list as input, and outputs the vector of motion in terms of a turn and distance to move.

To evaluate performance we ran two experiments. The objective of the first experiment was to form a hexagon. The desired distance R between robots was 23 inches. Using the theory, we chose a G of 270 ($p = 2$ and $F_{max} = 1$). The beginning configuration was random. The results were very consistent, producing a good quality hexagon ten times in a row and taking approximately seven cycles on average. A cycle takes about 25 seconds to perform, almost all of which is devoted to the scan of the environment. The AP algorithm itself is extremely fast. A new localization technology that we are developing will be much faster and will replace the current scan technique. For all runs the robots were separated by 20.5 to 26 inches in the final formation, which is only slightly more error than the sensor error.

The objective of the second experiment was to form a hexagon and then move in formation to a goal. For this experiment, we placed four photo-diode light sensors on each robot, one per side. These produced an additional force vector, moving the robots towards a light source (a window). The magnitude of the goal force must be less than $\sqrt{3}G/R^p$ for cohesion of the formation to be maintained [11]. The results, shown in Figure 6, were consistent over ten runs, achieving an accuracy comparable to the formation experiment above. The robots moved about one foot in 13 cycles of the AP algorithm.

In conclusion, the ability to set system parameters from theory greatly enhances our ability to generate correct robotic swarm behavior.

4 Discussion and Outlook

This paper has summarized our framework for distributed control of swarms of robots in sensor networks, based on laws of artificial physics (AP). The motivation for this approach is that natural laws of physics satisfy the requirements

```
void ap() {
  int theta, index = 0;
  float r, F, fx, fy, sum_fx = 0.0, sum_fy = 0.0;
  float vx, vy, delta_vx, delta_vy, delta_x, delta_y;
  vx = vy = 0.0; // Full friction.
  // Row i of robots[][] is for the ith robot located.
  // Column 0/1 has the bearing/range to that robot.
  while ((robots[index][0] != -1)) { // For all neighboring robots do:
    theta = robots[index][0];        // get the robot bearing
    r = robots[index][1];            // and distance.
    if (r > 1.5 * R) F = 0.0; // If robot too far, ignore it.
    else {
        F = G / (r * r);   // Force law, with p = 2.
        if (F > F_MAX) F = F_MAX;
        if (r < R) F = -F; // Has effect of negating force vector.
    }
    fx = F * cos(theta);   // Compute x component of force.
    fy = F * sin(theta);   // Compute y component of force.
    sum_fx += fx;          // Sum x components of force.
    sum_fy += fy;          // Sum y components of force.
    index++;
  }
  delta_vx = delta_T*sum_fx; // Change in x component of velocity.
  delta_vy = delta_T*sum_fy; // Change in y component of velocity.
  vx = vx + delta_vx;        // New x component of velocity.
  vy = vy + delta_vy;        // New y component of velocity.
  delta_x = delta_T*vx;      // Change in x component of position.
  delta_y = delta_T*vy;      // Change in y component of position.
  // Distance to move.
  distance = (int)sqrt(delta_x*delta_x + delta_y*delta_y);
  // Bearing of movement.
  turn = (int)(atan2(delta_y, delta_x));
  // Turn robot in minimal direction.
  if (delta_x < 0.0) turn += 180; }
```

Fig. 5. The main AP code, which takes as input a robot neighbor list (with distance and bearing information) and outputs a vector of motion.

Fig. 6. Seven robots self-organize into a hexagonal formation, which then successfully moves towards a light source (a window, not the reflection of the window). Pictures taken at the initial conditions, at two minutes, fifteen minutes, and thirty minutes.

of distributed control, namely, self-organization, fault-tolerance, and self-repair. The results have been quite encouraging. We illustrated how AP can self-organize hexagonal and square lattices. Results showing fault-tolerance and self-repair are in [1]. We have also summarized simulation results with dynamic multi-agent behaviors such as obstacle avoidance, surveillance, and sweeping. This paper also outlines several physics-based analyses of AP, focusing on potential energy, force balance equations, and kinetic theory. These analyses provide a predictive technique for setting parameters in the robotic systems. Finally, we have shown AP on a team of seven mobile robots.

We consider AP to be one level of a more complex control architecture. The lowest level controls the movement of the robots. AP is at the next higher level, providing "way points" for the robots, as well as providing simple repair mechanisms. Our goal is to put as much behavior as possible into this level, in order to provide the ability to generate laws governing important parameters. However, levels above AP are needed to solve more complex tasks requiring planning, learning, and global information [12].

5 Future Work

Currently, we are improving our mechanism for robot localization. This work is an extension of Navarro-Serment et. al. [13], using a combination of RF with acoustic pulses to perform trilateration. This will distinguish robots from obstacles in a straightforward fashion, and will be much faster than our current "scan" technique.

We also plan to address the topic of optimality, if needed. It is well understood that *potential field* (PF) approaches can yield sub-optimal solutions. Since AP is similar to PF, similar problems arise with AP. Our experience thus far indicates that this is not a crucial concern, especially for the tasks that we have examined. However, if optimality is required we can apply new results from control theory to design force laws that guarantee optimality [14,15]. Although oscillations of the formations do not occur, excess movement of the robots can occur due the fact that the force law $F = Gm_i m_j / r^p$ is not zero at the desired separation distance R. Current work using an alternative force law based on the Lennard-Jones potential, where the magnitude of the force is negligible at the desired separation, greatly minimizes this motion.

From a theoretical standpoint, we plan to formally analyze other important aspects of AP systems. This analysis will be more dynamic (e.g., kinetic theory) than the analysis presented here. We also intend to expand the repertoire of formations, both static and dynamic. For example, initial progress has been made on developing static and dynamic linear formations. Many other formations are possible within the AP framework. Using evolutionary algorithms to create desired force laws is one intriguing possibility that we are currently investigating. We summarize one preliminary experiment here.

5.1 Evolving Force Laws for Surveillance

This task consists of an environment with areas of forest and non-forest. The goal is for a swarm of MAVs to locate tanks on the ground. Tanks are hidden from the MAVs if they are in the forest. Each MAV has a target sensor with a small field of view for locating the tanks (with probability of detection P_d), and a foliage sensor with a larger field of view for detecting forest below it. The environment is shown in Figure 7 with three MAVs. The smallest circle represents the target sensor. The next largest circle represents the foliage field of view. Each MAV acts as if it were contained in a "bubble" that has a certain radius (depicted as the outer circle). If the bubbles of two MAVs are separated from each other, the MAVs are attracted to one another. If the bubbles overlap, they are repelled. The optimum MAV separation occurs when the bubbles touch.

A genetic algorithm is used to find the optimum bubble radius, as well as the G, p, and F_{max} parameters of the force law. We generated one environment with 100 tanks, 25% forest coverage, 20 MAVs, and $P_d = 1$. The GA fitness function was the percentage of tanks seen within 3000 time steps. In this "training" phase the GA was used to evolve a force law, that when used by all MAVs, created perfect coverage (all tanks were seen).

Testing consisted of generating other environments and performing ablation studies. First, nine other environments were created with the same parameter settings. The MAVs had no difficulty finding all tanks. Next, the percentage of foliage was systematically changed from 0% to 90% in increments of 10%. In all cases the MAVs found all tanks. Finally, two ablation studies were performed.

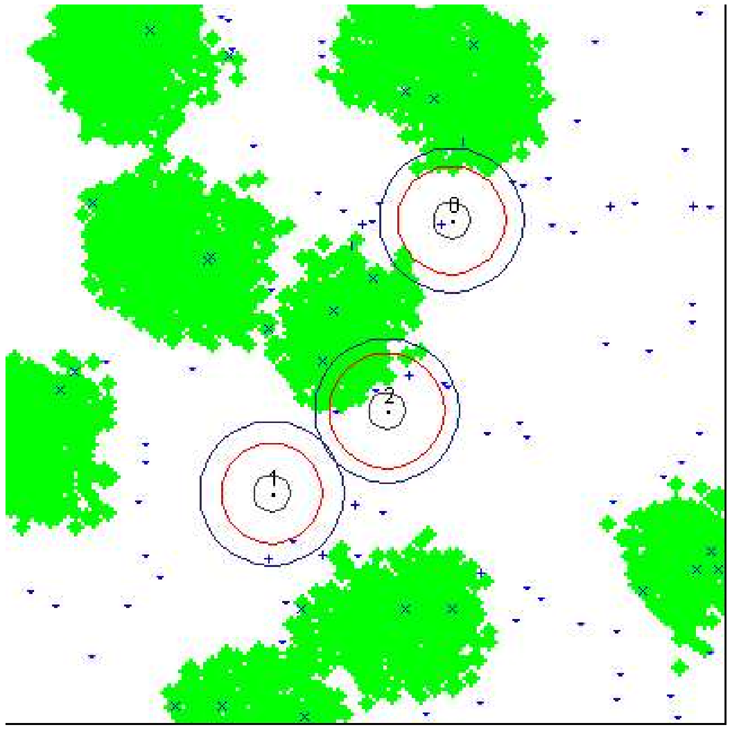

Fig. 7. The surveillance environment, showing areas of forest, three MAVs, and 100 tanks. The triangle represents a tank that has not yet been seen but is visible, the + represents a tank that has been seen and is visible, the × represents a hidden tank that has not been seen, and the | represents a tank that is currently hidden but has been previously seen.

First, the number of MAVs was reduced from 20, to 15, to 10, and then to 5. The results were quite robust; performance only suffered when the number of agents was reduced to 5. Second, we also lowered the probability of detection P_d from 1.0, to 0.75, to 0.5, and then to 0.25. Again, the results were quite robust, showing negligible performance drops (see Figure 8).

In summary, the results are extremely promising. Using only one training environment, the GA evolved a force law that showed surprising generality over changes in the environment, the number of MAVs, and the quality of the target detection sensor.

6 Related Work

Most of the swarm literature can be subdivided into *swarm intelligence*, *behavior-based*, *rule-based*, *control-theoretic* and *physics-based* techniques. Swarm intelligence techniques are ethologically motivated and have had excellent success

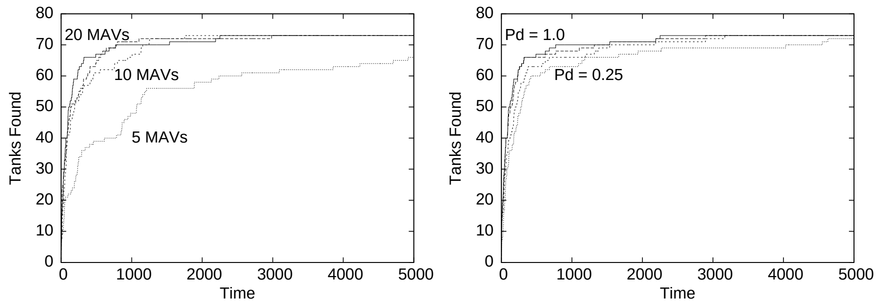

Fig. 8. The number of tanks found as the number of MAVs is reduced (left graph). The number of tanks found as the probability of detection (P_d) of the target sensor is reduced (right graph). The number of visible tanks is 73.

with foraging, task allocation, and division of labor problems [16]. In Beni et. al. [17,18], a swarm distribution is determined via a system of linear equations describing difference equations with periodic boundary conditions. Behavior-based approaches [19,20,21,22] are also very popular. They derive vector information in a fashion similar to AP. Furthermore, particular behaviors such as "aggregation" and "dispersion" are similar to the attractive and repulsive forces in AP. Both behavior-based and rule-based (e.g., [23]) systems have proved quite successful in demonstrating a variety of behaviors in a heuristic manner. Behavior-based and rule-based techniques do not make use of potential fields or forces. Instead, they deal directly with velocity vectors and heuristics for changing those vectors (although the term "potential field" is often used in the behavior-based literature, it generally refers to a field that differs from the strict Newtonian physics definition). Control-theoretic approaches have also been applied effectively [14]. Our approach does not make the assumption of having leaders and followers [24].

One of the earliest physics-based techniques is the potential fields (PF) approach (e.g., [8]). Most of the PF literature deals with a small number of robots (typically just one) that navigate through a field of obstacles to get to a target location. The environment, rather than the agents, exert forces. Obstacles exert repulsive forces, while goals exert attractive forces. Recently, Howard et al. [2] and Vail and Veloso [25] extended PF to include inter-agent repulsive forces – for the purpose of achieving coverage. Although this work was developed independently of AP, it affirms the feasibility of a physics force-based approach. The *social potential fields* [26] framework by Reif and Wang is highly related to AP, in that they rely on a force-law simulation similar to our own. We plan to merge their approach with ours.

7 Acknowledgments

The surveillance task mentioned in this paper is supported by DARPA grant #DODARMY41700. Thanks to Vaibhav Mutha for an early version of the obstacle avoidance code.

References

1. Spears, W., Gordon, D.: Using artificial physics to control agents. In: IEEE International Conference on Information, Intelligence, and Systems, Washington, DC (1999) 281–288
2. Howard, A., Matarić, M., Sukhatme, G.: Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In: Sixth International Symposium on Distributed Autonomous Robotics Systems, Fukuoka, Japan, ACM (2002) 299–308
3. Gordon-Spears, D., Spears, W.: Analysis of a phase transition in a physics-based multiagent system. In Hinchey, M., Rash, J., Truszkowski, W., Rouff, C., Gordon-Spears, D., eds.: Lecture Notes in Computer Science. Volume 2699., Greenbelt, MD, Springer-Verlag (2003) 193–207
4. Kellogg, J., Bovais, C., Foch, R., McFarlane, H., Sullivan, C., Dahlburg, J., Gardner, J., Ramamurti, R., Gordon-Spears, D., Hartley, R., Kamgar-Parsi, B., Pipitone, F., Spears, W., Sciambi, A., Srull, D.: The NRL micro tactical expendable (MITE) air vehicle. The Aeronautical Journal **106** (2002) 431–441
5. Zarzhitsky, D., Spears, D., Thayer, D., Spears, W.: Agent-based chemical plume tracing using fluid dynamics. In Hinchey, M., Rash, J., Truszkowski, W., Rouff, C., eds.: Formal Approaches to Agent-Based Systems, Greenbelt, MD, Springer-Verlag (2005)
6. Spears, W., Spears, D., Hamann, J., Heil, R.: Distributed, physics-based control of swarms of vehicles. Autonomous Robots **17** (2004)
7. Spears, W., Spears, D.: A formal analysis of potential energy in a multiagent system. In Hinchey, M., Rash, J., Truszkowski, W., Rouff, C., eds.: Formal Approaches to Agent-Based Systems, Greenbelt, MD, Springer-Verlag (2005)
8. Khatib, O.: Real-time obstacle avoidance for manipulators and mobile robots. International Journal of Robotics Research **5** (1986) 90–98
9. Kerr, W., Spears, D., Spears, W., Thayer, D.: Two formal fluids models for multiagent sweeping and obstacle avoidance. In Hinchey, M., Rash, J., Truszkowski, W., Rouff, C., eds.: Formal Approaches to Agent-Based Systems, Greenbelt, MD, Springer-Verlag (2005)
10. Kerr, W., Spears, D., Spears, W., Thayer, D.: Swarm coverage through a partially obstructed corridor. (in preparation)
11. Spears, W., Heil, R., Spears, D., Zarzhitsky, D.: Physicomimetics for mobile robot formations. In: International Conference on Autonomous Agents and Multi Agent Systems. (2004)
12. Gordon, D., Spears, W., Sokolsky, O., Lee, I.: Distributed spatial control, global monitoring and steering of mobile physical agents. In: IEEE International Conference on Information, Intelligence, and Systems, Washington, DC (1999) 681–688
13. L. Navarro-Serment, L., Paredis, C., Khosla, P.: A beacon system for the localization of distributed robotic teams. In: International Conference on Field and Service Robots, Pittsburgh, PA (1999) 232–237

14. Fax, J., Murray, R.: Information flow and cooperative control of vehicle formations. In: IFAC World Congress, Barcelona, Spain (2002)
15. Olfati-Saber, R., Murray, R.: Distributed cooperative control of multiple vehicle formations using structural potential functions. In: IFAC World Congress, Barcelona, Spain (2002)
16. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, Santa Fe Institute Studies in the Sciences of Complexity, Oxford, U.K. (1999)
17. Beni., G., Hackwood, S.: Stationary waves in cyclic swarms. Intelligent Control (1992) 234–242
18. Beni, G., Wang, J.: Swarm intelligence. In: Proceedings of the Seventh Annual Meeting of the Robotics Society of Japan, Tokyo, Japan (1989) 425–428
19. Balch, T., Arkin, R.: Behavior-based formation control for multi-robot teams. In: IEEE Transactions on Robotics and Automata. Volume 14. (1998) 1–15
20. Balch, T., Hybinette, M.: Social potentials for scalable multirobot formations. In: IEEE Transactions on Robotics and Automata. (2000)
21. Matarić, M.: Designing and understanding adaptive group behavior. Technical report, CS Dept, Brandeis Univ. (1995)
22. Payton, D., Daily, M., Hoff, B., Howard, M., Lee, C.: Pheromone robotics. In: SPIE Symposium on Intelligence Systems and Manufacturing, Boston, MA (2000)
23. Wu, A., Schultz, A., Agah, A.: Evolving control for distributed micro air vehicles. In: IEEE Conference on Computational Intelligence in Robotics and Automation, Belgium (1999) 174–179
24. Desai, J., Ostrowski, J., Kumar, V.: Modeling and control of formations of nonholonomic mobile robots. IEEE Transactions on Robotics and Automation **17** (2001) 905–908
25. Vail, D., Veloso, M.: Multi-robot dynamic role assignment and coordination through shared potential fields. In Schultz, A., Parker, L., Schneider, F., eds.: Multi-Robot Systems, Hingham, MA, Kluwer (2003) 87–98
26. Reif, J., Wang, H.: Social potential fields: A distributed behavioral control for autonomous robots. In: Robotics and Autonomous Systems. Volume 27 (3). (1999) 171–194