Distributed, Physics-Based Control of Swarms of Vehicles

William M. Spears
Diana F. Spears
Department of Computer Science

Jerry C. Hamann
Rodney Heil
Department of Electrical and Computer Engineering
College of Engineering
University of Wyoming
Laramie, WY 82071
wspears@cs.uwyo.edu

Abstract

We introduce a framework, called "physicomimetics," that provides distributed control of large collections of mobile physical agents in sensor networks. The agents sense and react to virtual forces, which are motivated by natural physics laws. Thus, physicomimetics is founded upon solid scientific principles. Furthermore, this framework provides an effective basis for self-organization, fault-tolerance, and self-repair. Three primary factors distinguish our framework from others that are related: an emphasis on minimality (e.g., cost effectiveness of large numbers of agents implies a need for expendable platforms with few sensors), ease of implementation, and run-time efficiency. Examples are shown of how this framework has been applied to construct various regular geometric lattice configurations (distributed sensing grids), as well as dynamic behavior for perimeter defense and surveillance. Analyses are provided that facilitate system understanding and predictability, including both qualitative and quantitative analyses of potential energy and a system phase transition. Physicomimetics has been implemented both in simulation and on a team of seven mobile robots. Specifics of the robotic embodiment are presented in the paper.

Keywords: swarm robotics, physicomimetics, self-organization, fault-tolerance, predictability, formations.

1. Introduction

The focus of our research is to design and build rapidly deployable, scalable, adaptive, cost-effective, and robust networks of autonomous distributed vehicles. This combines sensing, computation and networking with mobility, thereby enabling deployment, self-organization, and reconfiguration of the multi-agent collective. Our objective is to provide a scientific, yet practical, approach to the design and analysis of aggregate sensor systems.

The general purpose for deploying tens to hundreds of such agents can be summarized as "volumetric control." Volumetric control means monitoring, detecting, tracking, reporting, and responding to environmental conditions within a specified physical region. This is done in a distributed manner by deploying numerous vehicles, each carrying one or more sensors, to collect, aggregate, and fuse distributed data into a tactical assessment. The result is enhanced situational awareness and the potential for rapid and appropriate response. Our objective is to design fully automated, coordinated, multi-agent sensor systems.

The team vehicles could vary widely in type, as well as size, e.g., from nanobots or micro-electromechanical systems (MEMS) to micro-air vehicles (MAVs) and micro-satellites. An agent's sensors perceive the world, including other agents, and an agent's effectors make changes to that agent and/or the world, including other agents. It is assumed that agents can only sense and affect nearby agents; 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 agents (self-organization), but we also require fault-tolerance, that is, the global behavior degrades very gradually if individual agents 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 presents a framework, called "physicomimetics" or "artificial physics" (AP), for distributed control. We use the term "artificial" (or virtual) because although we are motivated by natural physical forces, we are not restricted to them [45]. Although the forces are virtual, agents *act* as if they were real. Thus the agent's sensors

must see enough to allow it to compute the force to which it is reacting. The agent's effectors must allow it to respond to this perceived force.

We see 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 agents, since their sensors and effectors will necessarily be primitive. Second, since the approach is largely independent of the size and number of agents, the results scale well to larger agents and larger sets of agents.

Three primary emphases distinguish the AP framework from others that are related: minimality, ease of implementation, and run-time efficiency. First, AP formations are achieved with a minimal set of sensors and sensor information. The rationale for this emphasis is that it will: (1) reduce overall vehicle cost, (2) enable physical embodiment with small agents, and (3) increase vehicle stealthiness if sensing is active. Second, the paper presents theoretical results that translate directly into practical advice on how to set system parameters for desired swarm performance. This makes the robotic implementation straightforward. Third, AP is designed to be computationally efficient. Therefore, we avoid physics-based multi-agent algorithms such as [32], which compute potential fields and then transform to forces at run-time. Instead, AP computes forces only at run-time.

The paper is organized as follows. First, we present the general AP framework, which is currently based on Newtonian physics, but is extendible to other types of physics. Then, a sequence of examples shows how the framework has been applied to construct a variety of both static and dynamic multi-agent formations and behaviors. This includes regular geometric lattices for distributed sensing, as well as dynamic behaviors for surveillance and perimeter defense. Fault-tolerance and self-repair are addressed in the context of these applications. Theoretical analyses are provided that facilitate deeper system understanding and predictability, including qualitative and quantitative analyses of a system phase transition and of system potential energy. Then, details are provided regarding the physical implementation of AP on a team of seven robots with minimal sensing capabilities. We conclude with discussions of related and future work.

2. The Physicomimetics Framework

The basic AP framework is elegantly simple. In essence, virtual physics forces drive a multi-agent 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 ($\vec{F} = m\vec{a}$) simulation.

At an abstract level, AP treats agents as physical particles. This enables the framework to be embodied in vehicles ranging in size from nanobots to satellites. Particles exist in two or three dimensions and are point-masses. Each particle i has position $\vec{x}$ and velocity $\vec{v}$. 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 $\Delta\vec{x}$. The perturbation depends on the current velocity, i.e., $\Delta\vec{x} = \vec{v}\Delta t$. The velocity of each particle at each time step also changes by $\Delta\vec{v}$. The change in velocity is controlled by the force on the particle, i.e., $\Delta\vec{v} = \vec{F}\Delta t/m$, where m is the mass of that particle and $\vec{F}$ is the force on that particle.[1] 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 agent's velocity (independently modeled in the same fashion by Howard et al. [23]).

We require that AP map easily to physical hardware, and our model reflects this design philosophy. Particle mass allows our simulated robots to have momentum. Robots need not have the same mass. The frictional force allows us to model actual friction, whether it is unavoidable or deliberate. With full friction, the robots come to a complete stop between sensor readings and with no friction the robots continue to move as they sense. The time step Δt reflects the amount of time the robots need to perform their sensor readings. If the time step is small the robots get readings frequently, whereas if the time step is large readings are obtained infrequently. 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.

[1] F and v denote the magnitude of vectors $\vec{F}$ and $\vec{v}$.

Although our framework does not require them, our design philosophy reflects further real-world constraints. The first is that AP be as distributed as possible and the second is that we require as little information as possible. To this end, we assume that sensors are minimal in information content and that the sensors (passive and active) are extremely local in nature. Occasionally we have to rely on very small amounts of global information and/or control, but this is done as infrequently as possible. If real-world systems have a richer suite of information, we can take advantage of it, but we do not rely on that information for the system to function.

Due to the particle-like nature of our simulation, one important aspect of the real world is not modeled, namely, collisions of robots with other robots or objects in the environment. This was a deliberate design decision, since we wanted AP to be as platform independent as possible. Once a physical platform is selected, that aspect of the simulation must be modeled separately, and a lower-level algorithm is responsible for collision avoidance. For example, with small physical robots, gentle collisions can be tolerated and dealt with by using simple bumper sensors and routines. However, with MAVs, collisions must be avoided. The AP framework can avoid collisions through strong repulsive forces, but if additional guarantees are required then they must be modeled separately.

Also, we do not model the behavioral dynamics of the actual robot. Although our robots can stop and turn on a dime, other platforms, such as MAVs, may not have this capability. AP is an algorithm that determines "way points" for the physical platforms. Lower-level software is necessary to control the movement of the robots toward their desired locations.

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 few sections, for a variety of static and dynamic multi-agent configurations. Our implementation with robots is discussed in Section 7.

3. Hexagonal Lattices

The example considered in this section was originally inspired by an application that required a swarm of MAVs to form a hexagonal lattice, thus creating a distributed sensing grid [25]. Such lattices create a virtual antenna or synthetic aperture radar to improve radar image resolution.

3.1 Designing Hexagonal Lattices

Since MAVs (or other small agents such as nanobots) will have simple sensors and primitive CPUs, our goal was to provide the simplest control rules requiring minimal sensors and effectors. At first blush, creating hexagons appears to be somewhat complicated, requiring sensors that can calculate distance, the number of neighbors, their angles, etc. However, only distance and bearing information is required. To understand this, recall an old high-school geometry lesson in which six circles of radius R can be drawn on the perimeter of a central circle of radius R. Figure 1 illustrates this construction. If the particles (shown as small circular spots) are deposited at the intersections of the circles, they form a hexagon with a particle in the middle.

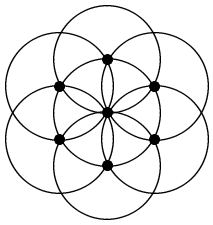

Figure 1. Six circles can be drawn on the perimeter of a central circle, forming a hexagon at the intersection of the circles.

The construction indicates that hexagons can be created via overlapping circles of radius R. To map this into a force law, imagine that each particle repels other particles that are closer than R, while attracting particles that are further than R in distance. Thus each particle has a circular "potential well" around itself at radius R – and neighboring particles 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 particles are likely to reside.

The nodes are the small circular spots in the previous figure. Thus the particles serve to create the very potential energy surface to which they are responding. Note that the potential energy surface is never actually computed by the robots. Robots only compute local force vectors. Potential energy is only computed for visualization or mathematical analysis.

With this in mind we defined a force law $F = Gm_i m_j / r^p$, 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. When $p = 0.0$ the force law is constant for all distances. Unless stated otherwise, we assume $p = 2.0$ and $F_{max} = 1$ in this paper. Also, $m_i = 1.0$ for all particles. 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. In a perfect hexagon, nearest neighbors are R away, and next nearest neighbors are $\sqrt{3}R$ away. Hence, particles have a visual range of only $1.5R$.

Figure 2 shows the magnitude of the force F, when $R = 50$, $G =$ 1,200, $p = 2$, and $F_{max} = 1$ (the system defaults). There are three discontinuities in the force law. The first occurs where the force law transitions from F_{max} to $F = Gm_i m_j / r^p$. The second occurs when the force law switches from repulsive to attractive at R. The third occurs at $1.5R\ (= 75)$, when the force goes to 0.

The initial conditions are also inspired by the MAV application. The MAVs are released from a canister dropped from a plane, then they propel outward (due to repulsive forces) until the desired geometric configuration is achieved. A two-dimensional Gaussian random variable (variance σ^2) initializes the positions of all particles. Their velocities are initialized to 0.0, although the framework does not require this. An example initial configuration for $N = 200$ particles is shown in Figure 3 (left). The 200 particles move for 1,000 time steps, using this very simple force law (see Figure 3, right). For $R = 50$, $G =$ 1,200 provides good results. These values remain fixed throughout this paper unless stated otherwise.

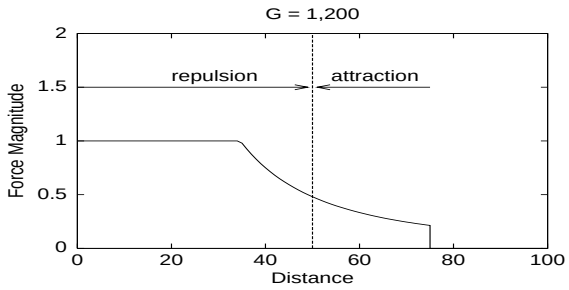

Figure 2. The force law, when $R = 50$, $G = 1{,}200$, $p = 2$ and $F_{max} = 1$. The force has a maximum magnitude of 1 and a magnitude of 0 at $1.5R = 75$. The force is repulsive when the distance is less than 50 and attractive when the distance is between 50 and 75.

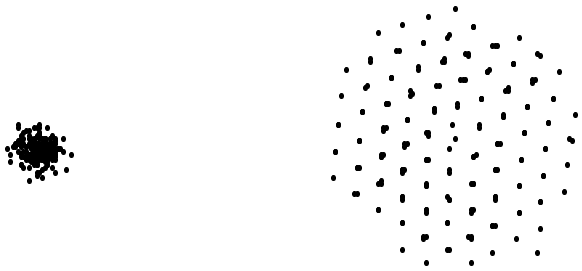

Figure 3. Initially, the particles are assumed to be in a tight cluster $t = 0$ (left). Then particles repel and after 1,000 time steps form a good hexagonal lattice (right).

There are a number of important observations to make about Figure 3 (right). First, a reasonably well-defined hexagonal lattice has been formed from the interaction of simple local force laws that involve only the detection of distance and bearing to nearby neighbors. The hexagonal lattice is not perfect – there is a flaw near the center of the structure. Also, the perimeter is not a hexagon, although this is not surprising, given the lack of global constraints. However, many hexagons are clearly embedded in the structure and the overall structure is quite hexagonal. Second, each node in the structure can have multiple particles ("a cluster"). Clustering is an emergent property that provides robustness, because the disappearance (failure) of individual particles from a cluster will have minimal effect. Clustering depends on the value of G, which we explore later in this section.

The formation shown in Figure 3 (right) is stable, and does not change to any significant degree as t increases past 1,000. The dynamics of the system ($t <$ 1,000) is fascinating to watch, yet is hard to simply convey in a paper.

As opposed to displaying numerous snapshots, we instead measure well-defined characteristics of the system at every time step. These characteristics provide useful insights into the system dynamics.

3.2 Evaluating Lattice Quality

The first characteristic we considered was orientation error, in the sense that the orientation of the lattice should be the same everywhere. To measure this characteristic, choose any pair of particles separated by $2R$. We use $2R$ instead of R to smooth out local noise, since we care about global error. Specifically, two particles are separated by $2R$ if $1.98R < r < 2.02R$. These two particles form a line segment. Then choose any other pair of particles separated by $2R$, forming another line segment. Measure the angle between the two line segments. For a hexagonal lattice, this angle should be close to some multiple of $60°$. The *error* is the absolute value of the difference between the angle and the closest multiple of 60. The maximum error is $30°$ and the minimum is $0°$. To evaluate lattice quality, we averaged the error over all distinct pairs of particle pairs.

Since error ranges from $0°$ to $30°$, the average error at $t = 0$ is $15°$. Then the error decreases – the rate at which the decrease occurs indicates how quickly the system is stabilizing. For this system, the error decreases smoothly until $t = 200$, resulting in a final error of $5.6°$ over the whole structure (averaged over 40 independent runs, with $\sigma = 3.6°$). Further improvements can be achieved by gradually reducing friction as time progresses.

3.3 Observing a Lattice Phase Transition

The second characteristic we considered was the size of clusters at the lattice nodes. For each particle i we counted the number of particles that were close to i ($0 < r < 0.2R$). We always included the particle i itself, so the minimum cluster size is 1.0. This was averaged over all particles and displayed for every time step (Figure 4, left). At $t = 0$ particles are close together, yielding a high clustering. The particles then separate, due to repulsion, so that by $t = 6$ the particles are apart. However, after $t = 6$ clusters re-emerge, with a final cluster size of roughly 2.5. The re-emergence of clusters serves to lower the total potential energy of the system, and the size

of the re-emerged clusters depends on G, R, and the geometry of the system. We summarize here one interesting experiment with G. We continued the previous experiment, evolving the system until t = 2,500. However, after t = 1,000, G is lowered by 0.5 at every time step. The results are shown in Figure 4 (right).

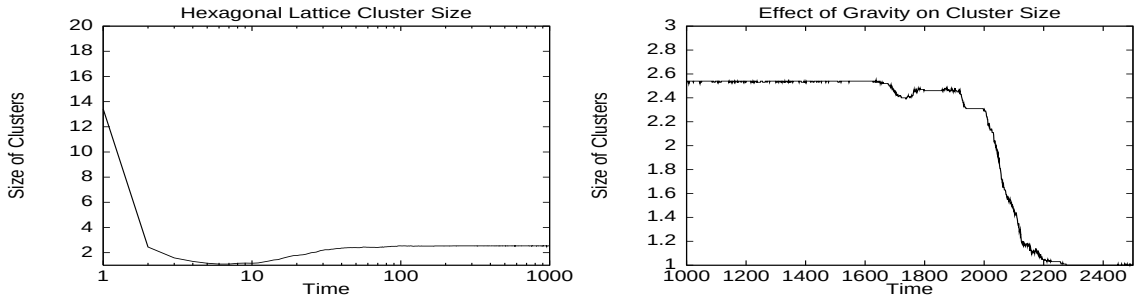

Figure 4. Initially, the particles have many nearby neighbors. Then the particles repel and separate at t = 6. As the hexagonal lattice forms, some particles form small clusters at the nodes of low potential energy t > 10 (left). After 1,000 time steps G is decreased linearly. At t = 2,200 (G = 700), the small clusters suddenly separate again, showing a phase transition (right). Note that the y-axis scale is different in the two graphs.

We expected the average cluster size to linearly decrease with G, but in fact the behavior was much more interesting. The cluster size remained relatively constant, until t = 2,000 (G = 700). At this point the cluster size dramatically dropped until t = 2,200 (G = 600), where the particles are separated (cluster size is one). This is very similar to a phase transition in natural physics, e.g., from a solid to a liquid.

3.4 Analysis of the Phase Transition

A primary objective of our research is to model and analyze multi-agent behavior, thus enabling predictions and, if needed, corrections. Complex, evolving multi-agent systems are notoriously difficult to predict. It is quite disconcerting when they exhibit anomalous behaviors for which there is no explanation. Because AP is built upon fundamental physics principles, it can be modeled and analyzed using traditional physics techniques, thus leading to explanatory physics-based laws.

Consider the case of the observed phase transition just described. We first conducted a *qualitative* analysis of this transition. In particular, we explored mathematical visualizations of the virtual potential energy (PE) fields. By definition, $V = -\int_s \vec{F} \bullet d\vec{s}$, where V is the traditional variable used for PE. This line (path) integral is a measure of the work done by a *virtual particle* to get to some position in the force field.

A line integral may be used to calculate PE if the force is (or is approximately) conservative because in that case the work to get from point a to point b is independent of the path taken. The force field is conservative if its curl (a measure of rotation of the vector force field near a point) is zero. Due to the radial symmetry of our force law, the curl is zero everywhere, and the PE field is meaningful.

Figure 5 (left) illustrates the PE field for a system of seven particles that have stabilized into a hexagonal formation (G = 1,200). Lighter shading represents high positive PE, while black represents low (zero or negative) PE. It is important to realize that the PE field shown does *not* illustrate the stability of the seven particles, because it is not from the point of view of any of those seven particles. Indeed, it is possible to construct such field diagrams, and they show that all seven particles are very stable. However, Figure 5 (left) illustrates what would happen if a *new* eighth virtual particle were brought into the system. Thus, the PE field is from the point of view of that virtual particle, and the PE at each (x, y) position is computed from the point of view of a virtual particle at that position. Positive PE indicates that work is required to push the virtual particle to that position. Negative PE indicates that work is required to push the virtual particle *away* from that position. A virtual particle placed in this field moves from regions of high PE to low PE. For example, consider the central particle, which is surrounded by a region of low PE. A virtual particle that is close to the central particle will stay near that center. Thus the central particle is in a PE well that can attract another particle. This is not surprising, since we showed earlier that a G of 1,200 results in clustering.

Now lower G to 800. Figure 5 (center) illustrates the PE field. While the central PE well is not as large as it was previously, it can still attract another particle. Finally, lower G to 600 (Figure 5, right). The central PE well no longer exists. The central particle is now surrounded by regions of lower PE – thus a virtual particle near the

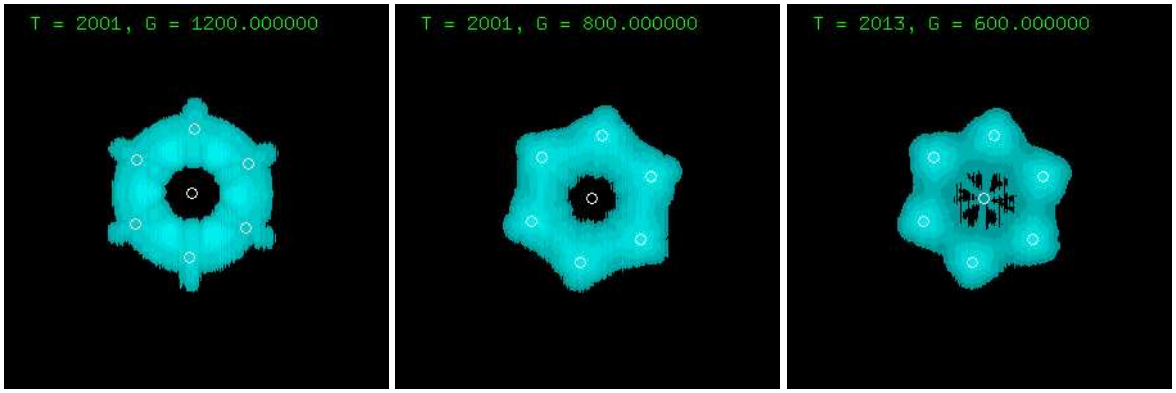

Figure 5. The PE field when $G = 1{,}200$ (left), $G = 800$ (middle), and $G = 600$ (right), for a system of seven particles. The PE field at each x, y position is computed from the point of view of a virtual particle at that position. Bright areas represent areas of positive potential energy, while black represents negative potential energy. A potential well surrounds the central particle for $G \geq 800$ but that well disappears when $G = 600$.

central particle will move away from that particle. A phase transition has occurred and clustering ceases.

Visualization of the PE field has led to an understanding of *why* the behavior of the dynamical system exhibits a phase transition. Large forces result in deep potential wells, allowing particles to form very stable sensing grids, with multiple particles clustering at nodes of low PE. In this situation the formation acts like a solid. The phase transition occurs when the potential wells disappear. At this point, the forces that promote cluster fragmentation are stronger than the forces that promote cluster cohesion. In this situation the formation acts like a liquid.

We have now set the stage for a *quantitative* analysis of the phase transition. In particular, we want to calculate the value of G (in terms of other parameter settings) where the phase transition will occur. Based on the qualitative analysis, we derived a standard *balance of forces* law to predict the phase transition. This quantitative law states that the phase transition will occur when the *cohesion force*, which keeps a particle within a cluster, equals the *fragmentation force*, which repels the particle from the cluster [19]. To specify this law, it is necessary to derive the expressions for these forces.

Figure 5 (right) indicates that a particle placed near the central particle will escape along trajectories that avoid

the perimeter particles. This has been confirmed via observation of the simulation. We depict these escape paths in Figure 6. In this figure, there are two particles at the center of the formation, and one particle each at the perimeter nodes. Label one of the two particles in the center as "A." Due to symmetry, without loss of generality we can focus on any of the escape paths for particle A. Let us examine the escape paths along the horizontal axis. Particle A can be expelled along this axis by the other central particle, which exerts a repulsive force of F_{max} (because r is small). Therefore, the fragmentation force upon particle A is F_{max}.

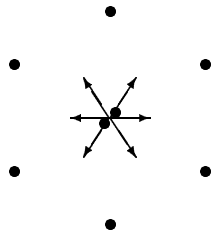

Figure 6. If two particles are at the center of a hexagon formation, one particle can escape along any of the six paths directed between the outer particles.

Next, we derive an expression for the cohesion force on A. Particle A is held near the center by the perimeter particles. Without loss of generality we again focus on the horizontal axis. Consider the force exerted by the four perimeter particles closest to the horizontal axis, on particle A. If A moves slightly to the right (or left), two particles will pull A back to the center (attraction), while two particles will push A back to the center (repulsion). All four particles contribute to the cohesion of the central cluster. For each particle, the magnitude of this force is G/R^p. The projection of this force on the horizontal axis is $\sqrt{3}/2$ times the magnitude of this force – because the angle between the chosen perimeter particles and the horizontal axis is 30°. Since there are four perimeter particles exerting this force (the remaining two have a force of 0 after projection), we multiply this amount by four to get a total cohesion force of $2\sqrt{3}G/R^p$.

When the cohesion force is greater than the fragmentation force, the central cluster remains intact. When the fragmentation force is greater, the central cluster separates. The phase transition occurs when the two forces are in balance: $F_{max} = 2\sqrt{3}G/R^p$. Thus the phase transition will occur when $G = F_{max}R^p/2\sqrt{3}$. We denote this

value of G as G_t. Thus our phase transition law is:

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

We tested this law for varying values of R, F_{max}, and p. The results are shown in Table 1, averaged over 10 independent runs, with $N = 200$. The system evolved until equilibrium with a high value of G. Then G was gradually lowered. Cluster size was monitored, and we noted the value of G when the average cluster size dropped below 1.5. The observed values are very close to those that are predicted (within 6%), despite the enormous range in the magnitude of predicted values (approximately four orders). The variance among runs is low, with the normalized standard deviation being less than 5.7%.

Table 1. The predicted/observed values of G_t for different values of R, p, and F_{max}. The three columns under F_{max} have $p = 2$. The three columns under p have $F_{max} = 1$. The predicted values are very close to those that are observed.

	F_{max}			p		
R	0.5	1.0	2.0	1.5	2.0	3.0
25	90/87	180/173	361/342	36/35	902/874	4,510/4,480
50	361/355	722/687	1,440/1,430	102/96	5,100/5,010	36,100/35,700
100	1,440/1,410	2,890/2,840	5,780/5,630	289/277	28,900/28,800	289,000/291,000

These results indicate that we have a very good predictor of G_t, which incorporates the most important system parameters p, R, and F_{max}. Notice that N (the number of particles) does not appear in our law. The phase transition behavior is largely unaffected by N, due to the local nature of the force law.

There are several uses for this equation. 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 $G \approx 0.9G_t$ will yield the best unclustered formations, while a value of $G \approx 1.8G_t$ will yield the best clustered formations. The reason for this is explored in the next section.

3.5 Conservation of Energy and the Role of Potential Energy

Because the force is conservative, AP should obey conservation of energy. Furthermore, as we shall see, the initial PE of the starting configuration yields important information concerning the dynamics of the system.

First, we measured the PE of the system at every time step, using the path integral shown earlier. This is the amount of work required to push each particle into position, one after another, for the current configuration of particles. Because the force is conservative, the order in which the particles are chosen is not relevant. Then we also measured the kinetic energy (KE) of the particles ($mv^2/2$). We modeled friction as heat energy. If there is no friction, the heat energy component is zero.

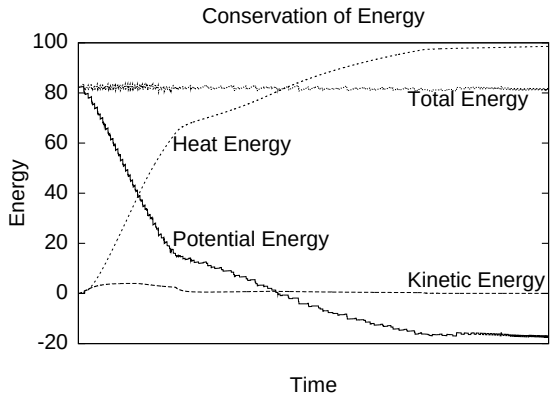

Figure 7. Conservation of energy, showing how the total energy remains constant, although the amount of different forms of energy changes over time. In the beginning, all energy is potential energy. This is transformed to kinetic energy when the particles move, and finally to heat as the particles stabilize due to friction.

Figure 7 illustrates the energy dynamics of AP. As expected, the total energy remains constant over time. The system starts with only PE. Note that the graph illustrates one of the foundational principles of AP, namely, that the system lowers PE until a minimum is reached. This reflects the stability of the final aggregate system, requiring work to move the system away from desired configurations (thus increasing PE).

As the system evolves, PE is converted into KE and heat, and the particles exhibit maximum motion (see Figure 7). Finally, the particles slow, and only heat remains. Note that PE is negative after a certain point. This

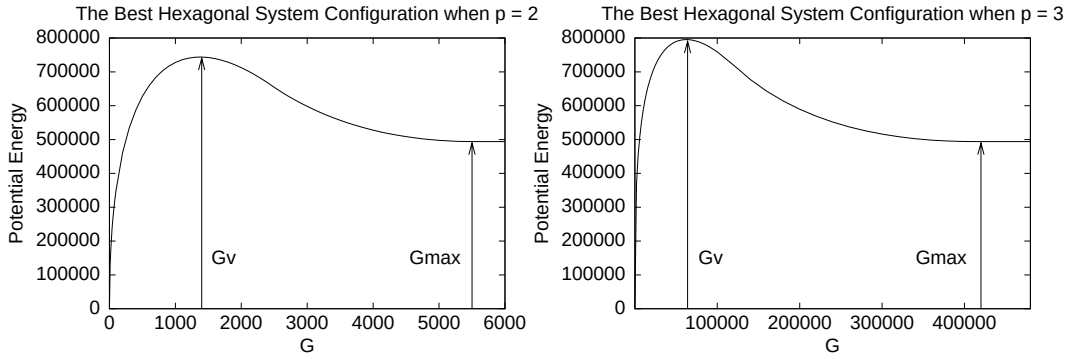

Figure 8. The amount of potential energy of the initial configuration of the hexagonal lattice system is maximized when G_V = 1,300 and G_V = 65,000, for a 200 particle system, when $p = 2$ (left) and $p = 3$ (right). The arrows show the values of G_V and G_{max}, where G_{max} is the maximum setting of G.

illustrates stability of individual particles (as well as the collective) – it would require work to push individual particles out of these configurations. Hence this graph shows that the system is resilient to moderate amounts of force acting to disrupt it, once stable configurations are achieved. This issue will be addressed in the next section.

The initial configuration ($t = 0$) PE also predicts important properties of the final evolved system, namely how well it evolves and the size of the formation. Higher initial PE determines that more work will be done by the system – and the creation of bigger formations requires more work. Higher initial PE is also correlated with better formations, because higher PE leads to greater initial linear momentum of the particles. As with simulated annealing, this momentum can help overcome problems with local optima.

For example, consider Figure 8 (left), which shows the PE of the initial configuration of the 200 particle system, when $p = 2$, for different values of G. In the graphs, G_V is the value of G at which PE is maximized, and G_{max} is the largest useful setting of G. Interestingly, PE is maximized at the range of values of G (1,200 – 1,400) that have been found empirically to yield the best structures. To test this hypothesis, we recalculated PE for the system when $p = 3$. The results are shown in Figure 8 (right). Again, maximum PE is achieved for a G value that is very close to those that yield the best structures.

As with the phase transition analysis, the goal is to derive a general expression for G_V. We first need to calculate

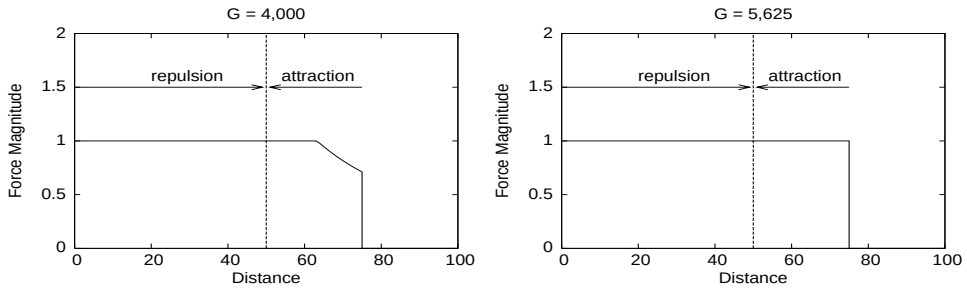

Figure 9. The force law, when G = 4,000 (left) and G = 5,625 (right), representing the second and third situations. The force has a maximum magnitude of 1 and a magnitude of 0 at $1.5R = 75$. The force is repulsive when the distance is less than 50 and attractive when the distance is between 50 and 75.

the potential energy, V. We begin by calculating the PE of a two particle system.

It is necessary to consider three different situations, depending on the radial extent to which F_{max} dominates the force law $F = G/r^p$. Recall that agents use F_{max} when $F \geq F_{max}$. This occurs when $G/r^p \geq F_{max}$ or, equivalently, when $r \leq (G/F_{max})^{1/p} \equiv R'$. The first situation occurs when F_{max} is used only at close distances, i.e., when $0 \leq R' \leq R$ (see Figure 2). The second situation occurs when $R \leq R' \leq 1.5R$. The third situation occurs when $R' > 1.5R$. In the third situation the force law has a constant magnitude of F_{max}, and V remains constant with increasing G (see Figure 9, left and right).

Let us now compute the PE for the first situation, which requires the calculation of three separate integrals. The first represents the attractive force felt by one particle as it approaches the other, from a distance of $1.5R$ to R. The second is the repulsive force of $F = G/r^p$ when $r < R$ and $F < F_{max}$. The third represents the repulsive force of F_{max} when $0 \leq r \leq R'$. Then:

$$V = -\int_R^{1.5R} \frac{G}{r^p}\, dr + \int_{R'}^{R} \frac{G}{r^p}\, dr + \int_0^{R'} F_{max}\, dr$$

The first term is negative because the force is attractive, whereas the latter two terms are positive because the force

is repulsive. We assume that $p \neq 1.0$, since AP is not run with that setting. Solving and substituting for R' yields:

$$V = \frac{[2R^{1-p} - (1.5R)^{1-p}]G}{(1-p)} - \frac{pG^{1/p}}{(1-p){F_{max}}^{(1-p)/p}}$$

The second situation is similar. The computation of PE is:

$$V = -\int_{R'}^{1.5R} \frac{G}{r^p}\, dr - \int_{R}^{R'} F_{max} dr + \int_{0}^{R} F_{max}\, dr$$

Solving and substituting for R' yields:

$$V = \frac{[(G/F_{max})^{(1-p)/p} - (1.5R)^{1-p}]G}{(1-p)} - F_{max}[2R - (G/F_{max})^{1/p}]$$

Finally, the third situation is:

$$V = -\int_{R}^{1.5R} F_{max} dr + \int_{0}^{R} F_{max} dr$$

Solving and substituting for R' yields:

$$V = \frac{F_{max} R}{2}$$

The first situation occurs with low G, when $G \leq F_{max} R^p$. The second situation occurs with higher values of G, when $F_{max} R^p \leq G \leq F_{max}(1.5R)^p$. The third situation occurs when $G \geq F_{max}(1.5R)^p$. In the third situation the PE of the system remains constant as G increases even further. Thus the maximum useful setting of G is $G_{max} = F_{max}(1.5R)^p$. We can see this in Figure 8 (which represent the full curves over all three situations) for values of $G_{max} =$ 5,625 and $G_{max} =$ 421,875 respectively. Above these values of G_{max}, PE stays constant.

It is now simple to generalize the computations for V to N particles, denoted as V_N. Regardless of the situation, we can build the N particle system one particle at a time, in any order (because forces are conservative), resulting

in an expression for the total initial PE:

$$V_N = \sum_{i=0}^{N-1} iV = \frac{VN(N-1)}{2}$$

where V is defined above for the two particle system.

Now that we have a general expression for the potential energy, V_N, we need to find the value of G that maximizes V_N. First, we need to determine whether the maximum occurs in the first or second situation. The slope of the PE equation for the second situation is strictly negative; thus the maximum must occur in the first situation. To find the maximum, we take the derivative of V_N for the first situation with respect to G, set it to zero, and solve for G. The resulting maximum is:

$$G_V = F_{max} R^p [2 - 1.5^{1-p}]^{p/(1-p)}$$

The value of G_V does not depend on the number of particles, which is a nice result. This simple formula is surprisingly predictive of the dynamics of a 200 particle system. For example, when $F_{max} = 1$, $R = 50$, and $p = 2$, G_V = 1,406, which is only about 7% higher than the value shown in Figure 8 (left). Similarly, when $p = 3$, G_V = 64,429, which is very close to the value shown in Figure 8 (right). The differences in values arise because our simulation has initial conditions specified by a 2D Gaussian random variable with a small variance σ^2, whereas our analysis assumes $\sigma^2 = 0$. Despite this difference, the equation for G_V works quite well.

In Section 3.4, empirical observations suggested that the best clustered formations occur when $G \approx 1.8 G_t$. This is equivalent to stating that $G_V / G_t \approx 1.8$, because maximum V is correlated with the best formations. Using our prior expressions for G_V and G_t, the ratio is:

$$G_V / G_t = 2\sqrt{3}[2 - 1.5^{1-p}]^{p/(1-p)}$$

The ratio depends only on p and the sensor range, but not on F_{max} or R. For $p = 2$ and $p = 3$ we get ratios of 1.9 and 1.77 respectively, which agree nicely with empirical observations.

Our final observation is that as G is increased beyond the optimal point G_V, PE decreases, yielding less energy to build large formations. As an example, for $p = 2$, with $N = 200$, we found empirically that when G = 1,200, the number of clusters in the final formation was 29. When G was doubled to 2,400 the number of clusters halved to 16. Finally, when G was doubled again to 4,800 the number of clusters was 7. In this situation G_{max} = 5,625. When G is set to the maximum, a minimal structure consisting of four clusters in a diamond formation is created. This result appears to hold in general, regardless of system parameters.

In summary, we have built a picture of how to set the value of G, given other system parameters. For unclustered behavior, set G to be slightly lower than the phase transition point G_t. For the best clustered behavior with the largest formations, set G to G_V (which is greater than G_t). For the smallest formations with maximal clustering, set G to G_{max}. We are currently attempting to predict the number of clusters given G.

3.6 Robustness

If $G \approx G_V$, and the system has reached equilibrium, then it is very robust with respect to the disappearance of numerous particles. Since lattice nodes of low PE are created via the intersection of many circular PE wells, the removal of particles from a node decreases the PE well depth of neighboring nodes but usually does not alter the lattice structure. The lattice is also preserved because non-neighboring nodes are unaffected by the particle removal – since they are out of sensing range. However, if enough particles disappear from a node, the balance of forces at neighboring nodes can change enough to cause particles in those neighbor nodes to move. In particular, if the cluster fragmentation force exceeds the cohesion force at a neighbor node, then one or more particles will be ejected from that cluster. Nevertheless, an ejected particle will move to another node of low PE. If this node was previously empty, then the movement just described will partially repair the lattice. In summary, the lattice structure degrades slowly, except for possible fragmentation into disjoint sets in very rare situations, or when a

very large percentage of particles are removed. Figure 10 shows this clearly. Beginning with 99 particles, 10 particles are removed, then another 20, and finally another 20. Removed particles are randomly chosen from the interior and perimeter of the lattice. The lattice is reduced in overall size, but its overall structure and integrity remain intact. The lines in this figure represent the force bonds between particles, and are useful for visualization.

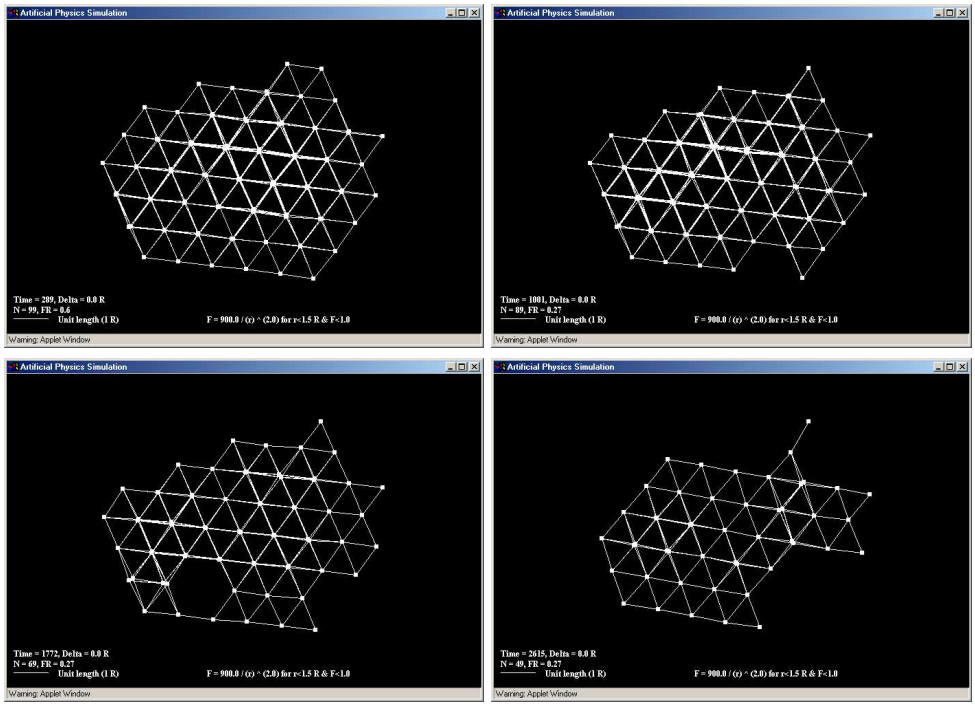

Figure 10. Beginning with 99 particles (top left), 10 particles are randomly removed (top right), then another 20 (bottom left), and finally another 20 (bottom right). The overall structure and integrity remain intact, demonstrating robustness.

The concept of PE also provides a natural mechanism for self-repair of formations if they are disturbed. The disturbances increase PE, and the system attempts to correct itself by lowering PE again. To test the efficacy of this approach we added a simulated blast (e.g., an explosion that causes a gust of wind) to our simulation. Weak gusts, which cause bends in the formation, are easily repaired with AP. More severe disturbances, that distort the shape of the perimeter, require monitoring, checking, and steering techniques [18].

4. Square Lattices

Given the success in creating hexagonal lattices, we investigated other regular structures. The square lattice is an obvious choice, since it also tiles a 2D plane.

4.1 Designing Square Lattices

The success of the hexagonal lattice hinged upon the fact that nearest neighbors are R in distance. This is not true for squares, since if the distance between particles along an edge is R, the distance along the diagonal is $\sqrt{2}R$. Particles have no way of knowing whether their relationship to neighbors is along an edge or along a diagonal.

Once again it appears that we need to know angles or the number of neighbors to solve this difficulty. However, a much simpler approach will do the trick. Suppose each particle is given another attribute, called "spin". Half of the particles are initialized to be spin "up", whereas the other half are spin "down".[2]

Figure 11. Square lattices can be formed by using particles of two "spins". Unlike spins are R apart while like spins are $\sqrt{2}R$ apart.

Consider the square depicted in Figure 11. Particles that are spin up are open circles, while particles that are spin down are filled circles. Particles of unlike spin are distance R from each other, whereas particles of like spin are distance $\sqrt{2}R$ from each other. This "coloring" of particles extends to square lattices, with alternating spins along the edges of squares, and same spins along the diagonals.

Figure 11 indicates that square lattices can be created if particles sense not only distance and bearing to neighbors, but also their spin. Thus sensors must detect one more bit of information, spin. We use the same force law as before: $F = Gm_im_j/r^p$. However, r is renormalized to $r/\sqrt{2}$ if two particles have the same spin. Once again the force is repulsive if $r < R$ and attractive if $r > R$. The one effector enables movement with velocity $v \leq V_{max}$.

[2]Spin is merely a particle label and has no relation to the rotational spin used in navigation templates [43].

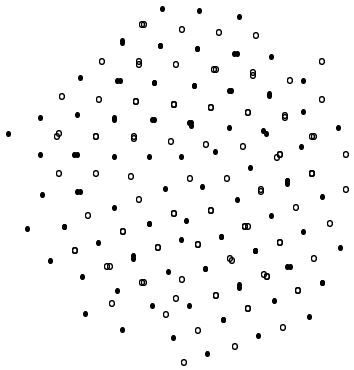

Figure 12. Using the same initial conditions as for the hexagonal lattice, 200 particles form a square lattice by $t = 4{,}000$, but global flaws exist.

To ensure that the force law is local, particles cannot see other particles that are further than cR, where $c = 1.3$ if particles have like spin and 1.7 otherwise.

The initial conditions are the same as those for the hexagonal lattice. The 200 particles move for 4,000 time steps (the system is somewhat slower to stabilize than the hexagon), using this very simple force law. The final result is shown in Figure 12. Again, we measure orientation error by choosing pairs of particle pairs separated by $2R$. By insisting that each particle pair has like spins, we ensure that pairs are aligned with the rows and columns of the lattice. In this case the angle between the two line segments should be close to some multiple of $90°$. The error is the absolute value of the difference between the angle and the closest multiple of 90. The maximum error is $45°$ while the minimum is $0°$. Averaged over 40 independent runs, the final error was about $12.8°$, with $\sigma = 6.7°$.

The results are clearly suboptimal. Locally, the particles have formed square lattices. This can be observed by noting that the spins alternate along the edges of squares, whereas spins are the same along diagonals. Once again each "node" in the lattice can have multiple particles, providing robustness (the average cluster size is roughly 1.75). However, large global flaws split the structure into separate square lattices. Thus, although the local force laws work reasonably well, they (not surprisingly) do not rule out difficulties at the global level. The question is whether global repair is needed or whether local repairs will suffice.

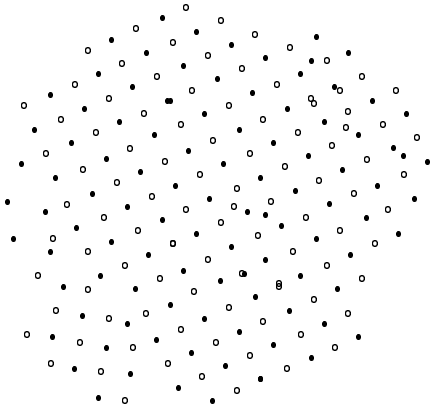

Figure 13. Using "spin-flip" local repair, the 200 particles form a better square lattice at $t = 4{,}000$. Global flaws are almost absent although some local flaws still exist.

4.2 Local Self-Repair of Square Lattices

As with other physical systems, noise can help remove global flaws in structures. Furthermore, systems should also self-repair at the local level. For example, if all particles at a particular lattice node are destroyed, a local hole opens in the lattice. Our goal is to provide a simple mechanism that repairs both local and global faults. To achieve this goal we focused again on the concept of spin. Figure 12 indicates that clusters are almost always made up of particles of like spin. There is an aversion to having clusters of unlike spins.

Spins are set at initialization. What would happen, though, if one particle in a cluster of like spins changes spin? It could fly away from that cluster to another cluster with the same spin as it now has. It could also land at an empty node which, although empty, is still an area of very low PE. In essence, clusters represent nodes with excess capacity, and that excess can fix problems in the structure as they arise. Our hypothesis is that this increased flow of particles (noise) can repair both local and global flaws in the square lattice.

Testing this hypothesis only required one change to the code. Again, particles are initialized with a given spin. However, if a particle has a very close neighbor ($r < 1.0$), the particle may flip its spin with a small probability. Particles have one additional effector – they can change their own spin. This does not create structural holes, since a particle only leaves a cluster if there is excess capacity in that cluster.

Once again, the 200 particles moved for 4,000 time steps, using the same force law, coupled with this simple spin-flip repair mechanism. The initial conditions were the same as those in the previous section. The results are shown in Figure 13. The previously shown global flaws are no longer in evidence, although a minor portion of the lattice is still misaligned. Many of the flaws that remain are local and are a result of a still operating spin-flip repair mechanism that continues to occasionally send particles from cluster to cluster. Observation of the evolving system shows that holes are continually filled, as particles leave their cluster and head toward open areas of low PE. An exact Wilcoxon rank-sum test indicates that the mean error *with* spin-flip repair ($4.9°$, $\sigma = 6.0°$) is statistically significantly less ($p < 0.001$) than the mean error *without* spin-flip repair ($12.8°$, $\sigma = 6.7°$).

4.3 Phase Transition Analysis

Square lattices also display a phase transition as G decreases. The derivation of a quantitative law for square lattices is a straightforward analogue of the analysis for hexagonal lattices. The one difference is that in a square lattice, one of the two particles in the central cluster is expelled along a path to one of the perimeter particles, rather than between them (see Figure 14).

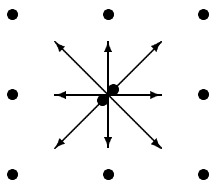

Figure 14. If two particles are at the center of a square formation, one particle can escape along any of the eight paths directed towards the outer particles.

In Figure 14, there are two particles in the center of the formation, and one particle each at the perimeter nodes. Label one of the two particles in the center as "A." Using the same reasoning as before the fragmentation force upon particle A is F_{max}. Particle A is held near the center by the perimeter particles. Using the geometry of the situation as we did with hexagons, the total cohesion force on A is $(2\sqrt{2}+2)G/R^p$ [19]. The phase transition

will occur when $G = F_{max}R^p/(2\sqrt{2}+2)$. The phase transition law for square lattices is:

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

Table 2. The predicted/observed values of G_t for different values of R, p, and F_{max}. The three columns under F_{max} have $p = 2$. The three columns under p have $F_{max} = 1$. The predicted values are very close to those that are observed.

	F_{max}			p		
R	0.5	1.0	2.0	1.5	2.0	3.0
25	65/69	130/136	259/278	26/26	647/651	3,236/3,312
50	259/272	519/530	1,036/1,066	73/74	3,662/3,730	25,891/26,850
100	1,036/1,112	2,071/2,138	4,143/4,405	207/206	20,713/21,375	207,125/211,350

We tested this law for varying values of R, F_{max}, and p. The results are shown in Table 2, averaged over 10 independent runs, with $N = 200$. The observed values are very close to those that are predicted (within 7%), and the normalized standard deviation is less than 6.2%.

4.4 Potential Energy Analysis

We can also compute the PE of the initial configuration. This computation is slightly more difficult than before because there are two "species" of particles (spin up and spin down), with different inter-species and intra-species sensor ranges. The computation is performed in three stages. First assemble all spin up particles together in a cluster. Then assemble all spin down particles in a cluster. Finally, join these two clusters together. We consider only the first situation ($0 \leq R' \equiv (G/F_{max})^{1/p} \leq R$), since this is where the maximum PE occurs.

First, compute the PE of the initial configuration of two spin up particles. When particles of like spin interact, r is renormalized by $\sqrt{2}$, and their sensor range is $1.3R$. Thus:

$$V = -\int_{\sqrt{2}R}^{1.3\sqrt{2}R} \frac{G}{(r/\sqrt{2})^p}\,dr + \int_{\sqrt{2}R'}^{\sqrt{2}R} \frac{G}{(r/\sqrt{2})^p}\,dr + \int_0^{\sqrt{2}R'} F_{max}\,dr$$

Solving and substituting for R' yields:

$$V = \sqrt{2}\left[\frac{(2R^{1-p} - (1.3R)^{1-p})G}{(1-p)} - \frac{pG^{1/p}}{(1-p){F_{max}}^{(1-p)/p}}\right]$$

The computation for V is very similar to that for the hexagonal lattice, differing only by a constant factor of $\sqrt{2}$ and the sensor range. We now generalize to N spin up particles:

$$V_N = \frac{VN(N-1)}{2}$$

The computation for spin down particles is identical. We now combine the two clusters of N spin up and N spin down particles:

$$V_{N+N} = V_N + V_N - \int_R^{1.7R} \frac{GN^2}{r^p}\,dr + \int_{R'}^{R} \frac{GN^2}{r^p}\,dr + \int_0^{R'} F_{max}N^2\,dr$$

Solving and substituting for R' yields:

$$V_{N+N} = V(N-1)N + N^2\left[\frac{(2R^{1-p} - (1.7R)^{1-p})G}{(1-p)} - \frac{pG^{1/p}}{(1-p){F_{max}}^{(1-p)/p}}\right]$$

To determine the value of G for which PE is maximized, we take the derivative of V_{N+N} with respect to G, set it to zero, and solve for G:

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

Note that in this case G_V depends on the number of particles N. It occurs because of the weighted average of different inter-species and intra-species sensor ranges. However, because this difference is not large, the dependency on N is also not large. For example, with $R = 50$, $F_{max} = 1$, and $p = 2$, then G_V = 1,466 if there

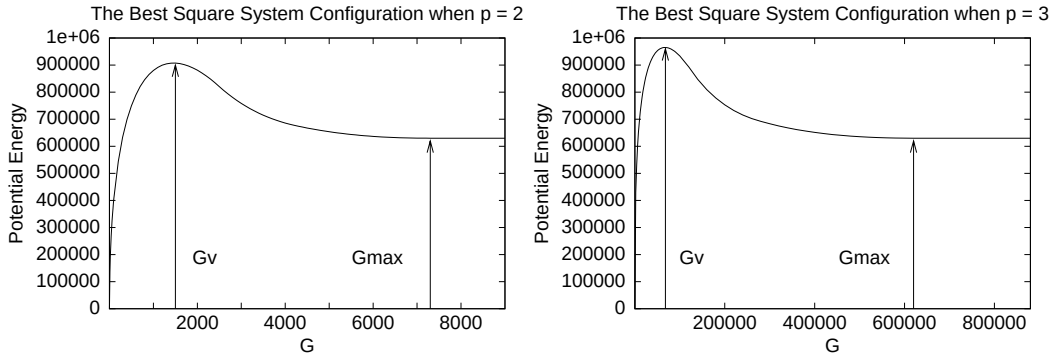

Figure 15. The amount of potential energy of the initial configuration of the square lattice system is maximized when G_V = 1,466 and G_V = 67,330, for a 200 particle system, when p = 2 (left) and p = 3 (right). The arrows show the values of G_V and G_{max}, where G_{max} is the maximum setting of G.

are 200 particles. With only 20 particles G_V = 1,456. Similarly, when $p = 3$, G_V = 67,330 and G_V = 66,960 respectively, for 200 and 20 particle systems (Figure 15).

As we did with the hexagonal lattices, we can also compute the value of G_{max}, which is the highest value of G that will have any effect on the system. For square lattices we get:

$$G_{max} = F_{max}(1.7R)^p$$

For our standard settings, when $p = 2$, $G_{max} = 7225$, and when $p = 3$, G_{max} = 614,125 (Figure 15).

5. Perfect Lattices and Transformations

Transformations are easily achieved in AP. For the original hexagonal and square lattices, transformations are accomplished by ignoring or not ignoring spins. Figure 16 illustrates a transformation from a square lattice to a hexagonal lattice, to another square lattice, and to a final hexagonal lattice. There is no guarantee that the same lattice will appear under successive transformations.

By adding other attributes [45], perfect hexagonal and square lattices (and their transformations) are also easily achieved. Each particle carries two different sets of attributes, one for hexagonal lattices and one for square lattices.

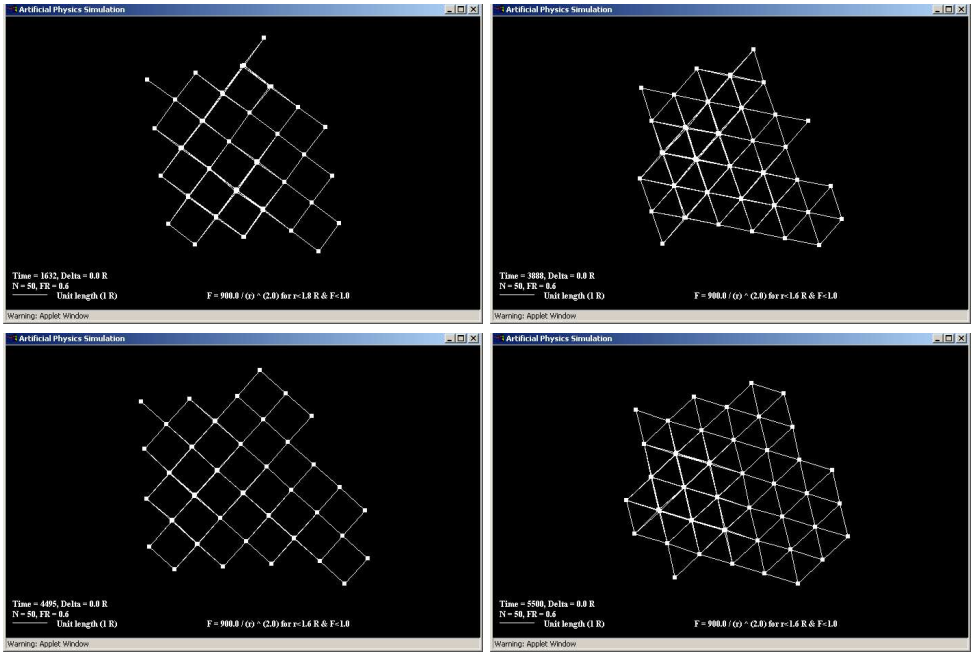

Figure 16. Agents can transform smoothly between hexagonal and square lattices. A square lattice (top left) transforms to a hexagonal lattice (top right), back to a square lattice (bottom left) and finally back to a hexagonal lattice (bottom right).

When the system switches from one pair to the other, it transforms. Figure 17 illustrates a transformation from a perfect square lattice to a perfect hexagonal lattice (given the number of particles). The square lattice structure is thoroughly destroyed, showing almost no structure at all. Despite this catastrophic disturbance the hexagonal structure eventually emerges, illustrating extreme robustness. The reverse transformation works equally well.

6. Other Formations in Two and Three Dimensions

Our simulation tool generalizes easily to three dimensions, which is necessary for the MAV task. Figure 18 shows 499 simulated MAVs in three separate planes of hexagonal lattices. Both top-down and side views are shown. Cubic lattices are also relatively easy to form, as are regular structures with triangular facets (such as triangular pyramids or regular icosahedrons). We have observed, however, as with natural crystals, that it is often easier to build these structures particle by particle, as opposed to building them in "batch."

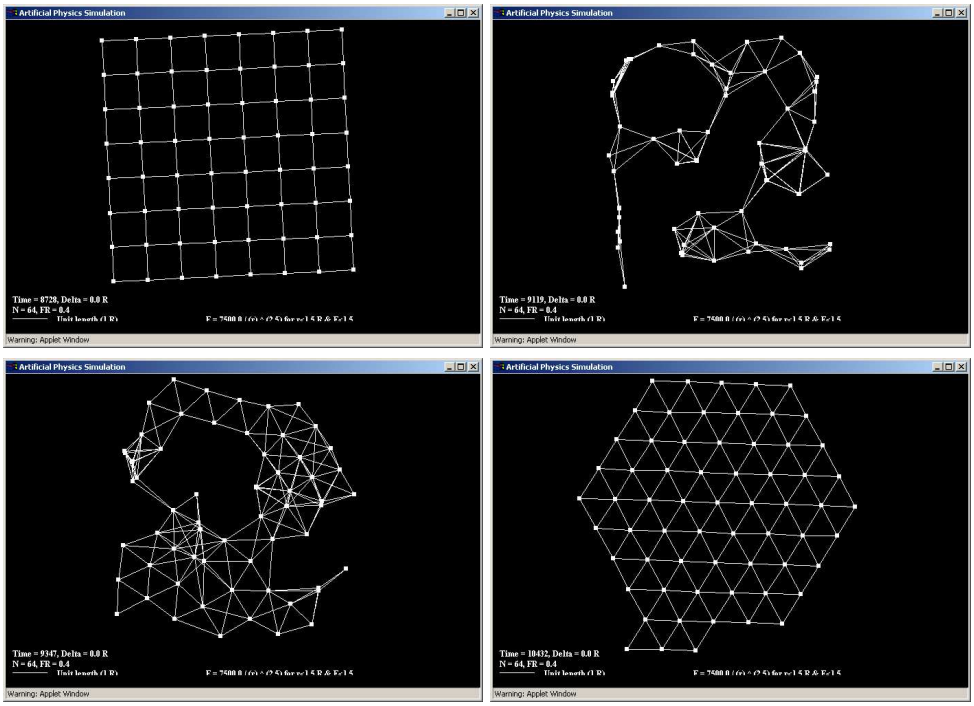

Figure 17. Agents can also transform from perfect square lattices to perfect hexagonal lattices (shown) and back (not shown).

The previous sections have described formations that have been planned in advance. However, our simulation tool provides the opportunity to change force law parameters in arbitrary and unusual ways. The results are often surprising, yielding unanticipated structures, especially in two dimensions. Figure 19 shows two unusual but potentially useful structures. We have found that some structures can be assembled easily by fixing the relevant parameters at the beginning. Others, however, are easier to create dynamically via transformation, as parameters are slowly changed. This also raises the possibility of searching the space of force laws (e.g., with genetic algorithms [22, 17, 44]) to create desired behavior.

7. Dynamic Behaviors: Obstacle Avoidance

Previous sections of this paper have focused on the creation of formations, but for most applications formations will have to move (often toward some goal). Also, for ground platforms, obstacles pose a serious challenge to the

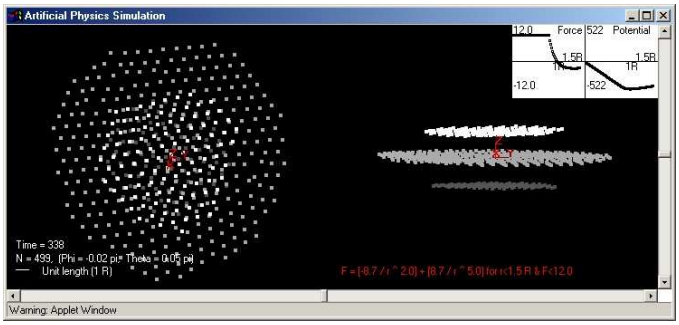

Figure 18. Three planes of MAVs in hexagonal lattices, shown from top-down and side views. There are 499 particles in this simulation.

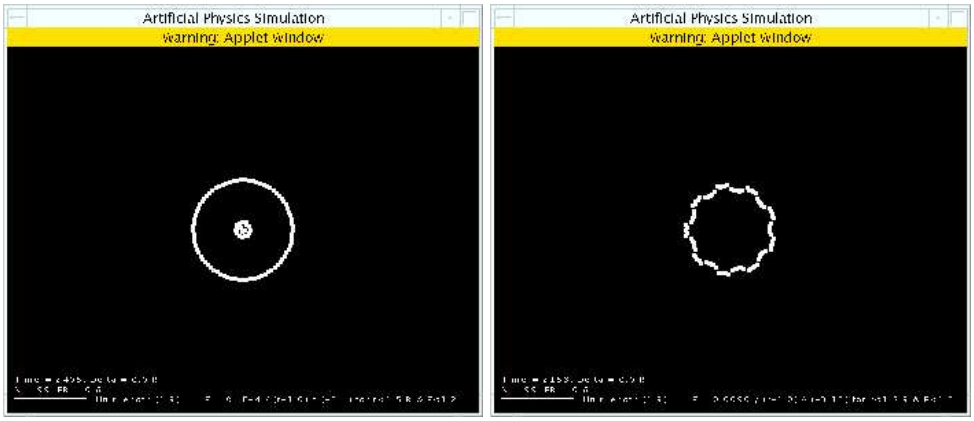

Figure 19. Unusual two dimensional formations can be achieved by changing the system parameters. These particular formations could be useful for perimeter defense applications.

movement of the formation. To address this, we have extended our simulation to include goals (both stationary and moving) as well as obstacles. Larger obstacles are created from multiple, point-sized obstacles; this enables flexible creation of obstacles of arbitrary size and shape. While the obstacles are stationary, the goal can be moved by the mouse as the simulation is running.

As a generalization to our standard paradigm, goals are attractive, whereas obstacles are repulsive (similar to potential field approaches, e.g., [26]). The goal can be sensed at a far distance of $6R$, while obstacles are sensed at a very short distance of $0.25R$. If the goal and obstacle forces are constant, we achieve good results.

Using this generalized paradigm we ran two sets of experiments. In the first, $G < G_t$, to remove clustering and to have fluid-like particle flow. The particles flow naturally around the obstacles, and do not retain any particular formation. In the second, $G \approx G_V > G_t$, to enhance the creation of clusters and rigid formations. Over numerous runs, three types of behavior are observed. First, the formation avoids obstacles via a sequence of rotations and counter-rotations of the whole collective. If this cannot be accomplished, the formation deforms by stretching force bonds between particles, diverging around obstacles directly in their path, and then converging again into a cohesive formation after the obstacle is passed. In the third situation, force bonds are broken and the formation fragments around an obstacle and then re-coalesces. One danger with this third situation is that particles can be permanently separated from the main formation.

In the previous experiments the goal was stationary. Preliminary results indicate that slowly moving goals are successfully tracked without difficulty.[3] The low G (liquid) version generally performs better on this task. One can easily imagine a situation where the formation lowers G to move around obstacles, and then raises G to achieve better formations after the obstacles have been avoided.

It is important to emphasize that this simulation models obstacle avoidance at an abstract level, and any application to platforms with complex dynamics (such as MAVs) will require additional modeling.

8. Dynamic Behaviors: Surveillance and Perimeter Defense

We have also explored surveillance and perimeter defense tasks. By using an analogy with the motion of gas molecules in a container, AP is successful on both of these tasks. The algorithm for surveillance is simple and elegant – agents repel each other, and are also repelled by the perimeter boundary. Friction is negligible. The surveillance task is shown in Figure 20 (left). Particles start at the center and move toward the perimeter, due to repulsion. They act like a gas in a container. If particles are destroyed, the remaining particles still search the enclosed area, but with less virtual "pressure." Likewise, the addition of particles is also treated gracefully,

[3] "Slowly" is relative to Δt. In the real world, faster motion implies smaller values of Δt (i.e., sensing must occur more often).

increasing the pressure in the system. The two small squares inside the perimeter represent intruders. Particles are attracted towards intruders, but since they also repel each other, the number of particles that can cover an intruder is limited.

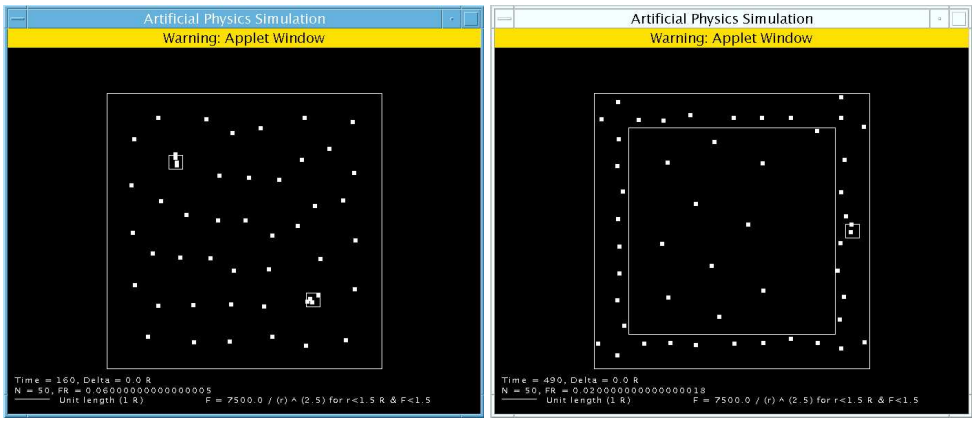

Figure 20. Simulated agents perform surveillance (left) and perimeter defense (right). For surveillance, agents are repelled by each other and by walls, while they are attracted by objects of interest. For perimeter defense agents repel each other and are drawn to the corridor between the two walls (the inner wall is porous and excess capacity is stored in the interior area). Friction is zero, because constant movement is required.

Perimeter defense is hardly more complex (see Figure 20, right). Once again, particles start from the center and repel each other. The inner and outer squares form a corridor to be monitored by the particles. The inner square is porous to the particles. Both the inner and outer walls are attractive, and particles are drawn to the space between them. One intruder is shown, which is attractive. Notice that some particles remain in the central area. This represents a situation of over-capacity – there are too many particles for the corridor. If particles in the corridor die, particles in the central area move to the corridor to replace them. This is a nice demonstration of the robustness of the system. An interesting phase transition of 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. Analysis of these dynamic systems will

center around the kinetic theory of gases, as has been initiated by Jantz et. al. [24].

9 Application to a Team of Mobile Robots

The current focus of this project is the physical embodiment of AP on a team of robots. Our choice of robots and sensors expresses a preference for minimal expense and expendable platforms.

For our initial experiments we have used inexpensive kits from KIPR (KISS Institute for Practical Robotics). These kits come with a variety of sensors and effectors, and two micro-computers – the RCX and the Handy Board. Due to its generality and ease of programming, we are currently using the Handy Board. The Handy Board has a HC11 Motorola processor with 32K of static RAM, a two line LCD screen, and the capacity to drive several DC motors and servos. It also has ports for a variety of digital and analog sensors.

Our robotic platform has two independent drive trains and two casters, allowing the platform to turn on a dime and move forward and backward. Slot sensors are incorporated into the drive trains to function as shaft encoders, yielding reasonably precise measures of the angle turned by the robot and the distance moved. The transmissions are geared down 25:1 to minimize slippage with the floor surface.

The "head" of the robot is a sensor platform used to detect other robots in the vicinity. For distance information we use Sharp GP2D12 IR sensors. This sensor provides fairly accurate distance readings (10% error over a range of 6 to 50 inches). The readings are relatively non-influenced by the material sensed, unless the material is highly reflective. However, the angle of orientation of the sensed object does have significant effects, especially if the object is reflective. As a consequence, each "head" is a circular cardboard (non-reflective) cylinder, allowing for accurate readings by the 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 (chair legs and walls). This algorithm detects nearby robots,

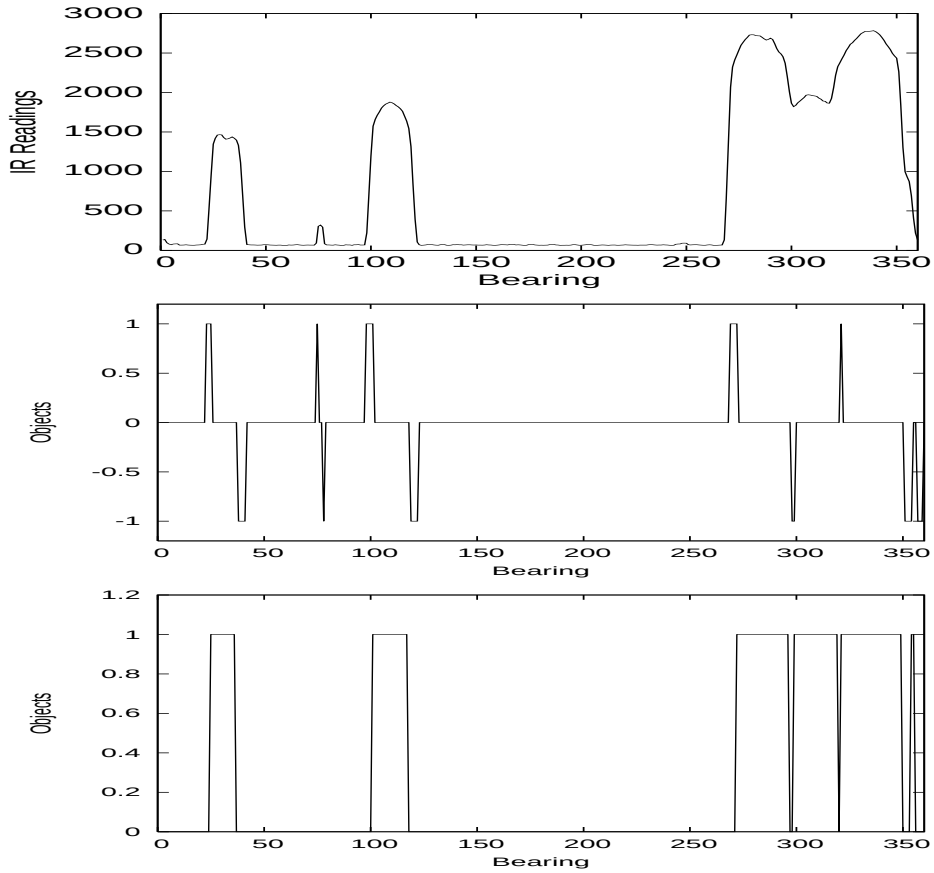

Figure 21. Example of robot detection where there are five nearby robots, one partially occluded by two others. The 360° scan produces a graph of distance values (top). The first derivative filter looks for large positive or negative values of the derivative, which yield object boundaries (middle). Regions between boundaries are potential objects. Objects that are too wide or are really empty space are filtered, producing an object list (bottom). The narrow false object to the right in the object list is also filtered.

producing a "robot" list that gives the bearing and distance to each neighboring robot (Figure 21).

Once sensing and object detection is 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 22 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.

For our experiments, we built seven robots. Each robot ran the same piece of software. The objective of the first experiment was to form a hexagon. The desired distance R between robots was 23 inches. Using the theory,

```
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; // Set x and y components of velocity to 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 = (int)sqrt(delta_x*delta_x + delta_y*delta_y); // Distance to move.
  turn = (int)(atan(delta_y / delta_x)); // Bearing of movement.
  if (delta_x < 0.0) turn += 180; }      // Turn robot in minimal direction.
```

Figure 22. The main AP code, which takes as input a robot neighbor list (with distance and bearing information) and outputs a vector of motion.

we chose a G of 270 ($p = 2$ and $F_{max} = 1$). The beginning configuration was random. The results were very consistent, producing a hexagon ten times in a row and taking approximately seven cycles on average. Our scan algorithm takes about 22 seconds per cycle for this first implementation; however, a new localization technology we are developing will be much faster. 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

Figure 23. Seven robots self-organize into a hexagonal formation, which then moves towards a light source. Pictures taken at the initial conditions, at two minutes, fifteen minutes, and thirty minutes.

force vector, moving the robots towards a light source (a window). The reflection of the window on the floor is not noticed by the robots and is not the light source. The results, shown in Figure 23, 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.

10. Summary and Related Work

This paper has introduced a framework for distributed control of swarms of vehicles in sensor networks, based on laws of artificial physics (AP). The motivation for this approach is that natural laws of physics satisfy the requirements 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. The concept of spin-flip from natural physics was shown to be a useful repair mechanism for square lattices, if

no global information is available. Structures in three dimensions are easily achieved, as well as transformations between structures. We have also presented preliminary results with dynamic multi-agent behaviors such as goal tracking, obstacle avoidance, surveillance, and perimeter defense. Finally, we have shown our first embodiment of AP on a team of seven mobile robots.

This paper also presents a novel physics-based analysis of AP, focusing on potential energy and force balance equations. This analysis provides a predictive technique for setting important parameters in the system, enabling the user to create unclustered formations, large clustered formations, and minimal clustered formations. The unclustered formations act like liquids, whereas the clustered formations act like solids. This analysis combines the geometry of the formations with important parameters of the system, namely, G, R, p, F_{max}, and sensor range. The parameter N was also included, but it is of little relevance for our most important results. This is a nice feature, since one motivation for AP was scalability to large numbers of agents. Including other relevant parameters such as Δt, V_{max} and friction requires a more dynamic analysis – we are currently focusing on "kinetic theory."

In conclusion, AP, although simple and elegant, has shown itself to be adept in a wide range of sensor network applications. AP demonstrates the capabilities of self-organization, fault-tolerance, self-repair,and effectiveness in spite of minimal sensing capabilities. There is a straightforward amenity to theoretical analysis, thereby enabling predictions of the behavior of the multi-agent swarm, and providing ease of implementation on a team of robots. Due to the lack of computation of potential fields, it is also computationally efficient.

Our discussion of related work will first focus on swarms and then on their theoretical analyses. Early work on swarm robotics focused on central controllers. For example, Carlson et al. [7] investigated techniques for controlling swarms of micro-electromechanical agents with a global controller that imposes an external potential field that is sensed by the agents. Recently, there has been movement away from global controllers, due to the brittleness of such an approach. AP is a distributed, rather than global, control framework for swarm management, although global control can be incorporated, if desired [18].

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 with foraging, task allocation, and division of labor problems [5, 20]. In Beni et. al. [4, 3], a swarm distribution is determined via a system of linear equations describing difference equations with periodic boundary conditions. Behavior-based approaches [6, 30, 35, 15, 2, 41] 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., [40, 51]) 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 (e.g., [1, 11]). Our approach does not make the assumption of having leaders and followers, as in [10, 13, 9].

One of the earliest physics-based techniques is the *potential fields* (PF) approach (e.g., [26]). 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. [23] and Vail and Veloso [49] 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. Another physics-based method is the "Engineered Collective" work by Duncan at the University of New Mexico and Robinett at Sandia National Laboratory. Their technique has been applied to search-and-rescue and other related tasks [39]. Kraus et al. [32] converts a generic, goal-oriented problem into a PE problem, and then automatically derives the forces needed to solve the problem. The *social potential fields* [37] framework is highly related to AP. Reif and Wang [37] rely on a force-law simulation that is similar to our own, allowing different forces between different agents. Their emphasis is on synthesizing desired formations by designing graphs that have a unique PE embedding. We plan to merge

this approach with ours.

Other physics-based approaches of relevance include research in flocking and other biologically motivated behavior. Reynolds models the physics of each agent and uses behavior-based rules to control its motion. Central to his work is "velocity matching", wherein each agent attempts to match the average velocity of its neighbors. The primary emphasis is on flocking [38]. Tu and Terzopoulos provide a sophisticated model of the physics of fish, which are controlled by behavior-based rules. The emphasis is on "schooling" and velocity matching [48]. Vicsek provides a point particle approach, but uses velocity matching (with random fluctuations) and emphasizes biological behavior [50, 8, 21]. His work on "escape panic" utilizes an $\vec{F} = m\vec{a}$ model, but includes velocity matching [21]. Toner and Tu provide a point particle model, with sophisticated theory, but again emphasize velocity matching and flocking behavior [47]. These models are quite different from ours, since we impose no velocity matching condition. Also, their models do not obey standard conservation laws. Furthermore, we utilize minimal sensory information, whereas velocity matching requires the computation of relative velocity differences between neighbors, which is more complex than our model. Finally, our motivation is to control vehicles for distributed sensing tasks. We are especially interested in regular geometric formations. For moving formations, our goal is to provide precise control of the formation, rather than "life-like" behavior.

One can also divide the related literature by the form of theoretical analysis, both in terms of the goal of the analysis and the method. There are generally two goals: stability and convergence/correctness. Under stability is the work by Schoenwald et. al. [39], Fierro et al. [14], Olfati-Saber and Murray [33], Liu et al. [29], and Lerman [28]. The first three apply Lyapunov methods. Liu et al. use a geometric/topological approach, and Lerman uses differential equations to model system dynamics. Under convergence/correctness is the work by Suzuki [46], Parker [34], and Liu et al. [29]. Methods here include geometry, topology and graph theory. Other goals of theoretical analyses include time complexity [32], synthesis [37], prediction of movement cohesion [29], coalition size [28], number of instigators to switch strategies [31], and collision frequency [24].

Methods of analysis are also similarly diverse. We focus only on physics-based analyses of physics-based

swarm robotics systems. We know of four methods. The first are the Lyapunov analyses by Schoenwald et. al. [39], Fierro et al. [14], and Olfati-Saber and Murray [33]. The second is the kinetic gas theory by Jantz et. al. [24]. The third is the minimum energy analysis by Reif and Wang [37]. The fourth develops macro-level equations describing flocking as a fluid-like movement [47].

To the best of our knowledge, the only analyses mentioned above that can set system parameters are those of Lerman [28], Numaoka [31], and Toner and Tu [47]. The first two analyses are of behavior-based systems, while the latter is of a "velocity matching" particle system. The capability of being able to set system parameters based on theory has enormous practical value, in terms of ease of implementation. The research presented in this paper provides practical laws for setting system parameters, to achieve the desired behavior with actual robots.

11. Future Work

Currently, we are improving our mechanism for robot localization. This work is an extension of Navarro-Serment et. al. [27], 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.

From a theoretical standpoint, we plan to formally analyze *all* 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 also plan to address the topic of optimality, if needed. It is well understood that 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 [33, 11].

Finally, future work will focus on transitioning to real-world applications. For example, to transition to MAVs,

more attention will be given to the interaction between the platforms and the environment. We are currently modeling four different environmental interactions: (1) goals, (2) obstacles, (3) the effects of wind, and (4) the friction of the medium. However, a richer suite is required for accurate models, such as signal propagation loss, occlusions of MAVs by obstacles or other MAVs, and weather effects. Also, if "velocity matching" is eventually required, information about "facing" and the relative speed of neighbors must be observed via sophisticated sensors or obtained via communication within a common coordinate system. Analysis will be more difficult under these situations. The speed of MAVs does not appear to be a major issue – higher speed implies a need for smaller Δt (i.e., faster sensors). AP is able to indicate how fast sensors must be, and to set limits on the strength of obstacle and goal forces to maintain a formation.

We consider AP to be one level of a more complex control architecture. The lowest level controls the actual movement of the platforms. 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, the current AP paradigm will not solve more complex tasks, involving planning, learning, repair from more catastrophic events, and global information. For example, certain arrangements of obstacles (such as cul-de-sacs) will require the addition of memory and planning. Hence, even higher levels will be required [42]. Learning is especially interesting to us, and we would like to add it to AP. Learning is advantageous in the context of behavior-based [12, 16] and rule-based [40, 36] systems, but its value has not been explored in the context of a physics-based system. Planning is also important, to provide forward reasoning, as opposed to reactive responses.

12. Acknowledgements

We are grateful to Adam Chiambi for developing our sophisticated Java simulator, and to Paul Hansen for building our robots. Thanks to Alan Schultz for loaning us some of his robot kits from the Naval Research Laboratory. We also thank Dimitri Zarzhitski, Suranga Hettiarachchi, and Vaibhav Mutha, for their participation

and contributions to this project. Vaibhav designed the balance of forces for obstacle avoidance and goal-seeking. Finally we thank Derek Green, Nathan Horter, Christoph Jechlitschek, Nathan D. Jensen, Nadya Kuzmina, Glenn McLellan, Markus Petters, Brendon Reed, and Peter Weissbrod of the Fall 2002 AI class for their contributions to the initial robot designs and experiments.

References

[1] R. Alur, J. Esposito, M. Kim, J. Kumar, and I. Lee. Formal modeling and analysis of hybrid systems: A case study in multi-robot coordination. *Lecture Notes in Computer Science*, 1708:212–232, 1999.

[2] T. Balch and R. Arkin. Behavior-based formation control for multi-robot teams. *IEEE Transactions on Robotics and Automata.*, 14(6):1–15, 1998.

[3] G. Beni. and S. Hackwood. Stationary waves in cyclic swarms. *Intelligent Control*, pages 234–242, 1992.

[4] G. Beni and J. Wang. Swarm intelligence. In *Proceedings of the Seventh Annual Meeting of the Robotics Society of Japan*, pages 425–428, Tokyo, Japan, 1989.

[5] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, Santa Fe Institute Studies in the Sciences of Complexity, Oxford, NY, 1999.

[6] D. Brogan and J. Hodgins. Group behaviors for systems with significant dynamics. *Autonomous Robots*, 4:137–153, 1997.

[7] B. Carlson, V. Gupta, and T. Hogg. Controlling agents in smart matter with global constraints. In Eugene C. Freuder, editor, *AAAI-97 Workshop on Constraints and Agents - Technical Report WS-97-05*, 1997.

[8] A. Czirok, M. Vicsek, and T. Vicsek. Collective motion of organisms in three dimensions. *Physica A*, 264(299):299–304, 1999.

[9] J. Desai, J. Ostrowski, and V. Kumar. Modeling and control of formations of nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation*, 17(6):905–908, 2001.

[10] J. Desai, J. Ostrowski, and V.Kumar. Controlling formations of multiple mobile robots. In *IEEE International Conference on Robotics and Automation*, pages 2864–2869, Belgium, 1998.

[11] J. Fax and R. Murray. Information flow and cooperative control of vehicle formations. In *IFAC World Congress*, Barcelona, Spain, 2002.

[12] F. Fernandez and L. Parker. Learning in large cooperative multi-robot domains. *International Journal of Robotics and Automation*, 16(4):217–226, 2002.

[13] R. Fierro, C. Belta, J. Desai, and V. Kumar. On controlling aircraft formations. In *IEEE Conference on Decision and Control*, volume 2, pages 1065–1070, Orlando, Florida, 2001.

[14] R. Fierro, P. Song, A. Das, and V. Kumar. Cooperative control of robot formations. In R. Murphey and P. Pardalos, editors, *Cooperative Control and Optimization*, volume 66, pages 73–93, Hingham, MA, 2002. Kluwer Academic Press.

[15] J. Fredslund and M. Matarić. A general algorithm for robot formations using local sensing and minimal communication. *IEEE Transactions on Robotics and Automation*, 18(5):837–846, 2002.

[16] D. Goldberg and M. Matarić. Learning multiple models for reward maximization. In *Seventeenth International Conference on Machine Learning*, pages 319–326, Stanford, CA, 2000.

[17] David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.

[18] D. Gordon, W. Spears, O. Sokolsky, and I. Lee. Distributed spatial control, global monitoring and steering of mobile physical agents. In *IEEE International Conference on Information, Intelligence, and Systems*, pages 681–688, Washington, DC, 1999.

[19] D. Gordon-Spears and W. Spears. Analysis of a phase transition in a physics-based multiagent system. In M. Hinchey, J. Rash, W. Truszkowski, C. Rouff, and D. Gordon-Spears, editors, *Lecture Notes in Computer Science*, volume 2699, pages 193–207, Greenbelt, MD, 2003. Springer-Verlag.

[20] A. Hayes, A. Martinoli, and R. Goodman. Distributed odor source localization. *IEEE Sensors*, 2 (3):260–271, 2002.

[21] D. Helbing, I. Farkas, and T. Vicsek. Simulating dynamical features of escape panic. *Nature*, 407:487–490, 2000.

[22] J. Holland. *Adaptation in natural and artificial systems*. University of Michigan Press, Michigan, USA, 1975.

[23] A. Howard, M. Matarić, and G. Sukhatme. 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*, pages 299–308, Fukuoka, Japan, 2002. ACM.

[24] S. Jantz, K. Doty, J. Bagnell, and I. Zapata. Kinetics of robotics: The development of universal metrics in robotic swarms. In *Florida Conference on Recent Advances in Robotics*, Miami, Florida, 1997.

[25] J. Kellogg, C. Bovais, R. Foch, H. McFarlane, C. Sullivan, J. Dahlburg, J. Gardner, R. Ramamurti, D. Gordon-Spears, R. Hartley, B. Kamgar-Parsi, F. Pipitone, W. Spears, A. Sciambi, and D. Srull. The NRL micro tactical expendable (MITE) air vehicle. *The Aeronautical Journal*, 106(1062):431–441, 2002.

[26] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5(1):90–98, 1986.

[27] L. L. Navarro-Serment, C. Paredis, and P. Khosla. A beacon system for the localization of distributed robotic teams. In *International Conference on Field and Service Robots*, pages 232–237, Pittsburgh, PA, 1999.

[28] K. Lerman and A. Galstyan. A general methodology for mathematical analysis of multi-agent systems. Technical Report ISI-TR-529, USC Information Sciences, 2001.

[29] Y. Liu, K. Passino, and M. Polycarpou. Stability analysis of m-dimensional asynchronous swarms with a fixed communication topology. In *IEEE Transactions on Automatic Control*, volume 48, pages 76–95, 2003.

[30] M. Matarić. Designing and understanding adaptive group behavior. Technical report, CS Dept, Brandeis Univ., 1995.

[31] C. Numaoka. Phase transitions in instigated collective decision making. *Adaptive Behavior*, 3(2):185–222, 1995.

[32] S. Kraus O. Shehory and O. Yadgar. Emergent cooperative goal-satisfaction in large-scale automated-agent systems. *Artificial Intelligence*, 110:1–55, 1999.

[33] R. Olfati-Saber and R. Murray. Distributed cooperative control of multiple vehicle formations using structural potential functions. In *IFAC World Congress*, Barcelona, Spain, 2002.

[34] L. Parker. Alliance: An architecture for fault tolerant multi-robot cooperation. *IEEE Transactions on Robotics and Automation*, 14(2):220–240, 1998.

[35] L. Parker. Toward the automated synthesis of cooperative mobile robot teams. In *SPIE Mobile Robots XIII*, volume 3525, pages 82–93, Boston, MA, 1998.

[36] M. Potter, L. Meeden, and A. Schultz. Heterogeneity in the coevolved behaviors of mobile robots: The emergence of specialists. In *Seventh International Conference on Artificial Intelligence*, pages 1337–1343, Seattle, Washington, 2001. Morgan Kaufmann.

[37] J. Reif and H. Wang. Social potential fields: A distributed behavioral control for autonomous robots. In *Robotics and Autonomous Systems*, volume 27 (3), pages 171–194, 1999.

[38] C. Reynolds. Flocks, herds, and schools: A distributed behavioral model. In *Proceedings of SIGGRAPH'87*, volume 21(4), pages 25–34, New York, NY, 1987. ACM Computer Graphics.

[39] D. Schoenwald, J. Feddema, and F. Oppel. Decentralized control of a collective of autonomous robotic vehicles. In *American Control Conference*, pages 2087–2092, Arlington, VA, 2001.

[40] A. Schultz, J. Grefenstette, and W. Adams. Roboshepherd: Learning a complex behavior. In *Robotics and Manufacturing: Recent Trends in Research and Applications*, volume 6, pages 763–768, New York, 1996. ASME Press.

[41] A. Schultz and L. Parker, editors. *Multi-Robot Systems: From Swarms to Intelligent Automata*. Kluwer, Hingham, MA, 2002.

[42] R. Simmons, T. Smith, M. Dias, D. Goldberg, D. Hershberger, A. Stentz, and R. Zlot. A layered architecture for coordination of mobile robots. In A. Schultz and L. Parker, editors, *Multi-Agent Robot Systems: From Swarms to Intelligent Automata*, Hingham, MA, 2002. Kluwer.

[43] M. Slack. *Situationally Driven Local Navigation for Mobile Robots*. PhD thesis, Virginia Polytechnic, 1990.

[44] W. Spears, K. De Jong, T. Bäck, D. Fogel, and H. de Garis. An overview of evolutionary computation. In *European Conference on Machine Learning*, volume 667, pages 442–459, Austria, 1993. Springer Verlag.

[45] W. Spears and D. Gordon. Using artificial physics to control agents. In *IEEE International Conference on Information, Intelligence, and Systems*, pages 281–288, Washington, DC, 1999.

[46] I. Suzuki and M. Yamashita. Distributed anonymous mobile robots: Formation of geometric patterns. *SIAM Journal of Computation*, 28(4):1347–1363, 1999.

[47] J. Toner and Y. Tu. Flocks, herds, and schools: A quantitative theory of flocking. *Physical Review E*, 58(4):4828–4858, 1998.

[48] X. Tu and D. Terzopoulos. Artifical fishes: Physics, locomotion, perception, behavior. In *Proceedings of SIGGRAPH'94*, pages 43–50, Orlando, Florida, 1994. ACM Computer Graphics.

[49] D. Vail and M. Veloso. Multi-robot dynamic role assignment and coordination through shared potential fields. In A. Schultz, L. Parker, and F. Schneider, editors, *Multi-Robot Systems*, pages 87–98, Hingham, MA, 2003. Kluwer.

[50] T. Vicsek, A. Czirok, E. Ben-Jacob, I. Cohen, and O. Shocher. Novel type of phase transition in a system of self-driven particles. *Physics Review Letters*, 75(6):1226–1229, 1995.

[51] A. Wu, A. Schultz, and A. Agah. Evolving control for distributed micro air vehicles. In *IEEE Conference on Computational Intelligence in Robotics and Automation*, pages 174–179, Belgium, 1999.