Using Artificial Physics to Control Agents

William M. Spears and Diana F. Gordon
AI Center, Naval Research Laboratory
Washington, D.C. 20375
spears@aic.nrl.navy.mil

Abstract

We introduce a novel framework called "artificial physics", which provides distributed control of large collections of agents. The agents react to artificial forces that are motivated by natural physical laws. This framework provides an effective mechanism for achieving self-assembly, fault-tolerance, and self-repair. Examples are shown for various regular geometric configurations of agents. A further example demonstrates that self-assembly via distributed control can also perform distributed computation.

1. Introduction

The objective of this research is the distributed control of agents that range in scale from neurons, nanobots, or micro-electromechanical systems (MEMS) to micro-air vehicles (MAVs) and satellites e.g., see [4]. Agents can be physical or virtual (e.g., softbots), mobile or immobile. Agents generally have sensors and effectors. An agent's sensors perceive the world (including other agents) and an agent's effectors make changes to that agent or the world (including other agents). Often, agents can only sense and affect nearby agents; thus the problem is usually one of "local" control. Sometimes control is also guided by global constraints and interactions.

Of course, one of the biggest problems is that we often don't know how to create the proper control rules. Not only do we want the desired global behavior to emerge from the local interaction between agents (i.e., self-assembly or self-organization), but we also would like there to be some measure of fault-tolerance i.e., the global behavior degrades very gradually if individual agents are damaged. Self-repair is also desirable, where a damaged system repairs itself.

Self-assembly, fault-tolerance, and self-repair are precisely those principles exhibited by natural systems. Thus, many answers to the problems of distributed control may lie in the examination of the natural laws of physics.

A recent research thrust that is based on natural physics suggests even more strongly the close connection between physics and distributed control. This exciting research thrust is the development of alternative distributed forms of computing based on nature, such as quantum computing, molecular computing, and computing with DNA e.g., see [1, 5]. Such computing engines are a direct result of the natural laws of physics. In the natural world small entities (quantum bits, molecules, etc.) exert forces on other entities and respond to forces from other entities. Generally the only forces that matter are those from nearby entities, thus the computation is performed via so-called "local" interactions. However, sometimes the computations are also guided by global constraints and interactions.

Clearly the fields of natural distributed computation and distributed control are related. Both fields involve the study of large numbers of entities (or agents) undergoing changes (or performing changes) due to global constraints and local interactions from nearby entities. The main difference is in the forces that control the entities. The forces in natural distributed computing are tied directly to physical laws. The forces in distributed control stem from man-made rules.

This paper proposes a general framework for distributed control in which "artificial physics" (AP) forces control agents. We use the term "artificial" because although we are motivated by natural physical forces, we are not restricted to only natural physical forces. Clearly, the agents aren't really subject to real forces, but they can *act* as if the forces are real. Thus the agent's sensors must see enough to allow it to compute the forces to which it is reacting. The agent's effectors must allow it to respond to this perceived force.

We see several 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 can be achieved through simple local interactions. This is required for very small agents (such as neurons or nanobots), since their sensors and effectors will necessarily be primitive. Two, since the approach is largely independent of the size and number of

1. REPORT DATE **1999**	2. REPORT TYPE	3. DATES COVERED **00-00-1999 to 00-00-1999**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Using Artificial Physics to Control Agents	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Research Laboratory,Artificial Intelligence Center,Washington,DC,20375**	8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
IEEE International Conference on Information, Intelligence, and Systems, Oct 31 -Nov 3,1999. Bethesda, MD

14. ABSTRACT
We introduce a novel framework called "artificial physics", which provides distributed control of large collections of agents. The agents react to artificial forces that are motivated by natural physical laws. This framework provides an effective mechanism for achieving self-assembly, fault-tolerance, and self-repair. Examples are shown for various regular geometric configurations of agents. A further example demonstrates that self-assembly via distributed control can also perform distributed computation.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**8**	

agents, the results should scale well to larger agents and larger sets of agents. Finally, we believe that this approach will tighten the connection between control and computation, potentially yielding new insights into computation or yielding new computational algorithms.

2. Framework

The motivation for this work stems from a desire for swarms of micro-air vehicles (MAVs) to form various regular geometric configurations – thus we will focus on mobile physical agents. Our approach treats agents as physical particles, which could range in size from nanobots to satellites. A simple but realistic physical simulation of the particles' behavior was built. Particles exist in two dimensions (we see little difficulty in generalizing to three dimensions) and are considered to be point-masses. Each particle i has position $p = (x, y)$ and velocity $v = (v_x, v_y)$. We use a discrete-time approximation to the continuous behavior of the particles, with time-step Δt. At each time step, the position of each particle undergoes a perturbation Δp. The perturbation depends on the current velocity $\Delta p = v \Delta t$. The velocity of each particle at each time step also changes by Δv. The change in velocity is controlled by the force on the particle $\Delta v = F \Delta t / m$, where m is the mass of that particle and F is the force on that particle. A frictional force is included, for self-stabilization.

For MAVs, the initial conditions are similar to those of a "big bang" – the MAVs are assumed to be released from a canister dropped from a plane, then they spread outwards until a desired geometric configuration is obtained. This is simulated by using a two dimensional Gaussian random variable to initialize the positions of all particles (MAVs). Velocities of all particles are initialized to be 0.0, and masses are all 1.0 (although the framework does not require this). An example initial configuration for 200 particles is shown in Figure 1.

Figure 1. The initial universe at $t = 0$.

Given the initial conditions and some desired global behavior, then, we must define what sensors, effectors, and force F laws are required such that the desired behavior emerges. We explore this in the next few sections, for different geometric configurations.

3. Creating Hexagonal Lattices

The example considered here is that of a swarm of MAVs whose mission is to form a hexagonal lattice, which cre-

ates an effective sensing grid. Essentially, such a lattice will create a virtual antenna or synthetic aperture radar to improve the resolution of radar images. A virtual antenna is expected to be an important future application of MAVs. Currently, the technology for MAV swarms (and swarms of other micro-vehicles such as micro-satellites) is in the early research stage. Nevertheless we are developing the control software now so that we will be prepared.

Since MAVs (or other small agents such as nanobots) have simple sensors and primitive CPUs, our goal was to provide the simplest possible control rules that require minimal sensors and effectors. At first blush, creating hexagons would appear to be somewhat complicated, requiring sensors that can calculate range, the number of neighbors, their angles, etc. However, it turns out that only range 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 (the fact that this can be done with only a compass and straight-edge can be proven with Galois theory). Figure 2 illustrates this construction. If the particles (shown as small circular spots) are deposited at the intersections of the circles, they form a hexagon.
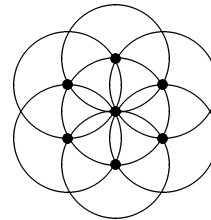

Figure 2. How circles can create hexagons.

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 can be considered to have a circular "potential well" around itself at radius R – neighboring particles will want to be at distance R from each other. The intersection of these potential wells is a form of constructive interference that creates "nodes" of very low potential energy where the particles will be likely to reside (again these are the small circular spots in the previous figure). Thus the particles serve to create the very potential energy surface they are responding to![1]

With this in mind we defined a force law $F = Gm_i m_j / r^2$, where F is the magnitude of the force between two particles i and j, and r is the range between the two particles. The "gravitational constant" G is set at initialization. The force is repulsive if $r < R$ and attractive if $r > R$. Each

---

[1]The entire potential energy surface is never actually computed. Particles only compute local force vectors for their current location.

particle has one sensor that can detect the range to nearby particles. The only effector is to be able to move with velocity v. To ensure that the force laws are local in nature, particles have a visual range of only $1.5R$.[2]

The initial universe of 200 particles (as shown in Figure 1) is now allowed to evolve for 1000 time steps, using this very simple force law (see Figure 3). For a radius R of 50 we have found that a gravitational constant of $G = 1200$ provides good results (these values remain fixed throughout this paper unless stated otherwise).
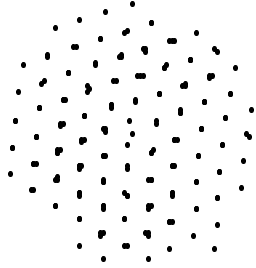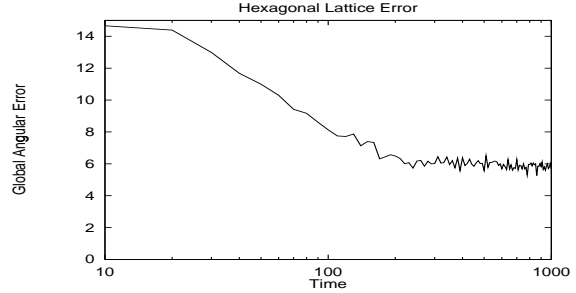

Figure 3. A good hexagonal lattice ($t = 1000$).

There are a number of important observations to make about Figure 3. First, it is obvious that a reasonably well-defined hexagonal lattice has been formed from the interaction of simple local force laws that involve only the detection of distance 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. The second observation is that each node in the structure can have multiple particles (i.e., multiple particles can "cluster" together). Clustering was an emergent property that we had not expected, and it provides increased robust behavior, because the disappearance (failure) of individual particles (agents) from a cluster will have minimal effect. This form of fault-tolerance is a result of the setting of G, which we explore later in this section.

The pattern of particles shown in Figure 3 is quite stable, and does not change to any significant degree as t increases past 1000. The dynamics of the evolving system (from $0 < t < 1000$) is quite fascinating (when watched on a computer screen), yet is hard to simply convey in a paper. As opposed to displaying numerous snapshots we have instead decided to graph certain well-defined characteristics of the system that can be measured at any time step. These characteristics yield useful insights into the system dynamics.

The first characteristic we examined is motivated by our

Figure 4. The average angular error in the structure as t increases. The log scale emphasizes early behavior.
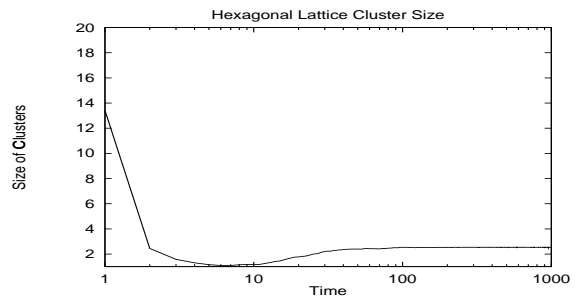

Figure 5. The size of clusters as t increases.

desire to have the global structure contain as few errors as possible, in the sense that the orientation of the hexagonal lattice should be the same everywhere throughout the lattice. To see how we can achieve a measure of this characteristic, consider choosing any pair of particles separated by $2R$. This forms a line segment. Then choose any other pair of particles also 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 $0°$. We averaged this over all distinct pairs of particle pairs, and displayed the average error for every ten time steps (see Figure 4).[3]

Since error ranges from $0°$ to $30°$, we expect the average error at the beginning to be around $15°$. After that the error should decrease – the rate at which the decrease occurs is a reasonable measure of how quickly the system is stabilizing. Error decreases smoothly until about $t = 200$, resulting in a final error of roughly $6°$ over the whole structure. This is a typical result. Averaged over 40 independent runs (different starting conditions) the final error was $5.6°$.

The second characteristic we examined is the size of

---

[2]The constant 1.5 is not chosen randomly. In a hexagon, if a nearby neighbor is further than R away, it is $\geq \sqrt{3}R$ away. We wanted the force laws to be as local as possible.

[3]We use $2R$ instead of R in an attempt to smooth out local noise, since we care about global error. A particle is considered to be separated by $2R$ if $1.98R < r < 2.02R$.

3
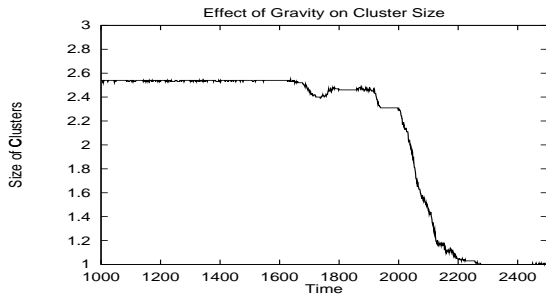
Figure 6. Cluster size drops suddenly as G is decreased linearly after $t = 1000$.

clusters. For each particle i we counted the number of particles that were close to i ($0 < r < 0.2R$). We always include the particle i itself, so the minimum size of a cluster is 1.0. This was averaged over all particles and displayed for every time step. Results are shown in Figure 5. At $t = 0$ all particles are very close to one another, yielding a high clustering. Immediately, the particles fly apart, due to the repulsive force, so that by $t = 6$ the particles are all effectively separated. However, after $t = 6$ clusters re-emerge, with the final cluster size being around 2.5. Clearly 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 factors such as G, R, and the geometry of the system. A full understanding of this phenomenon is beyond the scope of this paper, yet we summarize here one interesting experiment with G. We continued the previous experiment, evolving the system until $t = 2500$. However, after $t = 1000$ we lowered G by 0.5 for every time step. The results are shown in Figure 6.

We expected the average cluster size to linearly decrease with G, but in fact the behavior was much more interesting. The average cluster size remained quite constant, until about $t = 2000$, which is where G is 700. At this point the cluster size dramatically dropped until roughly $t = 2200$ (where $G = 600$), where the particles are separated again. This appears very similar to a phase transition in natural physics, demonstrating that AP can yield behavior very similar to that demonstrated in natural physics.

4. Creating Square Lattices

Given the success in creating hexagonal lattices, we were inspired to investigate other regular structures. Naturally the square lattice is an obvious choice, since (as with hexagons) squares will tile a 2D plane. The success of the hexagonal lattice hinged upon the fact that nearest neighbors are R in distance. Clearly this is not true for squares, since if the distance between particles along an edge is R, the distance

along the diagonal is $\sqrt{2}R$. The problem is that the particles have no way of knowing whether their relationship to neighbors is along an edge or along a diagonal.

Once again it would appear as if we would need to know angles or the number of neighbors to solve this difficulty. In fact, a much simpler approach will do the trick. Suppose that at creation each particle is given another attribute, called "spin". Half of the particles are initialized to be spin "up", while the other half are initialized to be spin "down". Spins do not change during the evolution of the system.[4]

Figure 7. Forming a square using two spins.

Consider the square depicted in Figure 7. Particles that are spin up are open circles, while particles that are spin down are filled circles. Note that particles of unlike spin are distance R from each other, while particles of like spin are distance $\sqrt{2}R$ from each other. This "coloring" of the particles extends to square lattices, with alternating spins along the edges of squares, and same spins along the diagonals.

The construction in Figure 7 indicates that square lattices can be created if particles can sense not only range to neighbors, but also the spins of their neighbors. Thus the sensors need to be able to detect one more bit of information, spin. We use the same force law as before: $F = Gm_im_j/r^2$. In this case, however, the range r is renormalized to be $r/\sqrt{2}$ if the two particles have the same spin. Then once again the force is repulsive if $r < R$ and attractive if $r > R$. The only effector is to be able to move with velocity v. To ensure that the force laws are local in nature, particles can not even see or respond to other particles that are further than $1.7R$.[5]

The initial universe of 200 particles is allowed to evolve for 4000 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 8. Again, we measure the angular error by choosing pairs of particle pairs separated by $2R$ (and by insisting that each particle pair have like spins, we help 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 $0°$. The graph of angular error is shown in Figure 9.

The results are clearly suboptimal. Locally, the particles have formed decent square lattices. This can be observed by noting that the spins alternate along the edges

---

[4] Spin is merely a particle label and has no relation to the rotational spin used in navigation templates [11].

[5] The constant is 1.7 if particles have like spin and 1.3 otherwise.
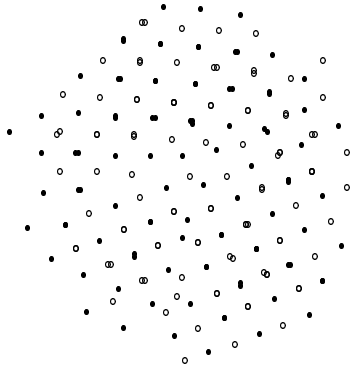
Figure 8. The 200 particles form a square lattice by $t = 4000$, but global flaws exist.
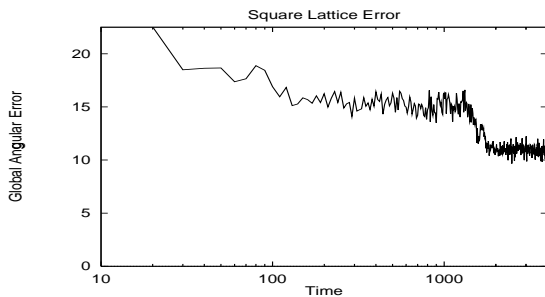
Figure 9. Angular error as t increases.

of squares, while spins are the same along the diagonal of squares. Once again each "node" in the lattice can have multiple particles, providing for increased robust behavior.[6] However, large global flaws split the structure into separate square lattices. This is also indicated by Figure 9, which shows that the system is unable to achieve an error of less than about $10°$. Again, this result is fairly typical. Averaged over 40 independent runs, the final error was about $12.8°$. Thus, although the local force laws do appear to work reasonably well, they (not surprisingly) do not rule out difficulties at the global level. The question is whether these sorts of difficulties must be repaired via global constraints or whether we can get by with local repairs.

5. Local Repair of Square Lattices

As with other physical systems, the presence of some form of noise often helps to remove global flaws in structures. Furthermore, we would also like systems to self-repair even at the local level. For example, if all particles at a particular node are destroyed, a local hole opens in the hexagonal or square lattice. Our goal is to provide a simple repair mechanism that repairs both local and global faults.

---

[6] In this case the final cluster size is roughly 1.75.

To achieve this goal we focused again on the concept of spin. If one examines Figure 8 one notices that clusters are almost always made up of particles of like spin. There is an aversion to having clusters of unlike spins (this was another surprising emergent property).

Now recall that spins are set at initialization and are not allowed to change. What would happen, though, if one particle in a cluster of like spins changes spin? It would probably 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 potential energy. In essence clusters represent areas with excess capacity (i.e., more than a sufficient number of particles), and that excess capacity can be used to fix problems in the structure as they arise. Thus our hypothesis is that this increased flow of particles (noise) can help repair both local and global flaws in the square lattice.

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

Once again the initial universe of 200 particles evolved using the same force law for the square lattice, coupled with this simple spin-flip repair mechanism. The initial conditions were the same as those in the previous section. The results are shown after 4000 time steps (see Figures 10 and 11) and are quite impressive. 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 flip spins (sending particles from cluster to cluster). Observation of the evolving system shows that holes are continually filled, as particles leave their cluster and head towards the open areas of low potential energy.

Note also that spin-flip repair has the effect of creating a larger square lattice pattern. This occurs because spin-flip repair will continually operate until each cluster contains only one particle. Note also that, as expected, Figure 11 shows increased noise, which is provided by the spin-flip repair. The noise allows a better global structure to emerge (the final error is $3.5°$).

To test our hypothesis that spin-flip repair serves to remove flaws from the evolved structure, we ran our system on the same 40 independent problems that were used with no spin-flip repair. Using an exact Wilcoxon rank-sum test we have determined that the mean error with spin-flip repair ($4.9°$) is statistically significantly less ($p < 0.001$) than the mean error without spin-flip repair ($12.8°$).
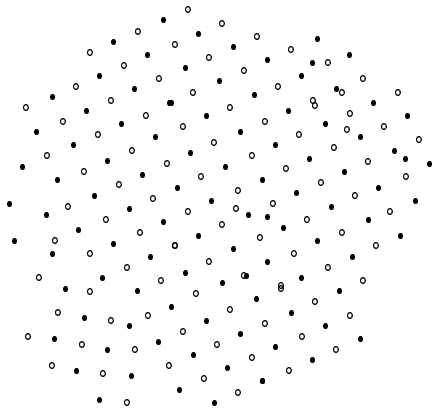
Figure 10. The 200 particles form a better square lattice at $t = 4000$. Global flaws are almost absent.
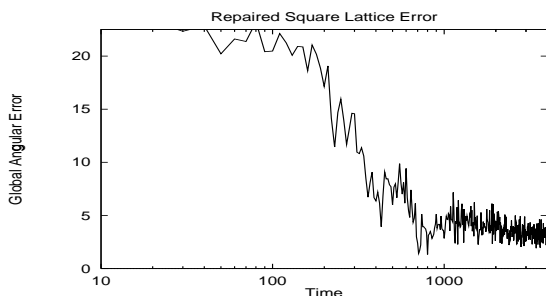

Figure 11. Angular error as t increases.

In this section we presented a repair mechanism for global and local flaws. Future work will introduce sensor and effector noise, as well as other faults. This will help us identify the limitations of our current approach, so that it can be improved as needed for greater self-repair ability.

6. Sorting

One motivation for AP stemmed from the connection between distributed control and natural distributed computation. Thus far we have shown that the framework of AP can be quite useful for distributed control. Can AP also perform distributed computation? To explore this possibility we considered the traditional computational task of sorting, which moves data from memory location to memory location, until the data is ordered. Suppose that each particle in the AP framework represented each datum. Could the particles physically move themselves so that they are aligned in the correct order? If so, we would have used distributed control to perform distributed computation!

We decided to investigate the task of 2D sorting along a square lattice. Interestingly, it turns out that the task can

be achieved using the above AP framework for constructing square lattices, with small modifications. Recall that we required the addition of a "spin" attribute for each particle, in order to construct square lattices. Every spin up (down) particle is indistinguishable from every other spin up (down) particle. However, for sorting every particle must be distinguishable. Thus we added a new 2D attribute (m, n) to each particle, where m and n are integers. Each particle receives a unique pair (m, n). The goal is to sort the particles according to m along the rows of the square lattice, while sorting the particles according to n along the columns.
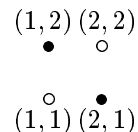

Figure 12. Four sorted particles.

Figure 12 represents a sorting of four particles that have the attributes (1,1), (1,2), (2,1) and (2,2). The astute reader will note that this sorting assumes an orientation to the system (i.e., that m should increase to the right, and n should increase upwards). The upshot is that local information is not sufficient for this task. One piece of global information is required, orientation. Thus particles must have sensors for determining range to other particles, their spin, (m, n) attribute, and orientation in the world.[7] It is important to note that a particle's (m, n) attribute does not represent a 2D coordinate, rather, it specifies its ordering relative to other particles. Thus the particle with attribute (2,1) should be to the right of (1,1).

We now use the same force law as before: $F = Gm_im_j/r^2$, where again the range r is renormalized to be $r/\sqrt{2}$ if the two particles have the same spin. Once again the force is repulsive if $r < R$ and attractive if $r > R$. However, there is now one further situation in which the force should be attractive, which is when two particles are not ordered properly with respect to their m or n attributes. The idea is that in this situation the two particles should be drawn to each other so that they pass by one another. Thus, the two particles will "dance" around one another, until their relative ordering is correct. Put another way, the force law now simultaneously enforces both a topology *and* a geometry to the system. The (m, n) attribute defines the topology, while spin and range define the geometry.

To test these ideas we chose a system of 225 particles, which were given the 225 attributes ranging from (1,1) to (15,15). For this task we do not want clustering, so we set $G = 600$ (as shown in our earlier results, this is sufficiently low to avoid the clustering effect). Spin-flip repair is not used. The system of 225 particles was randomly initialized in the same fashion as in the earlier experiments and
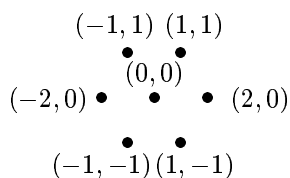
---

[7]A microscopic compass could sense orientation.
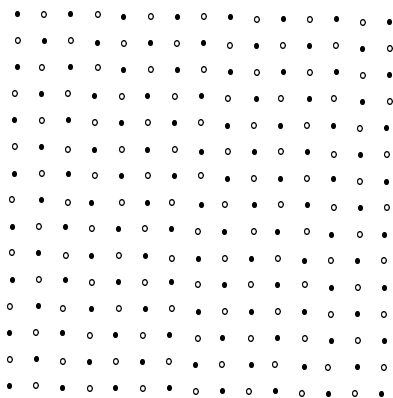
Figure 13. 225 particles form a perfect global square, using sorting ($t = 4000$).

Figure 15. 217 particles form a perfect global hexagon, using sorting ($t = 15000$).
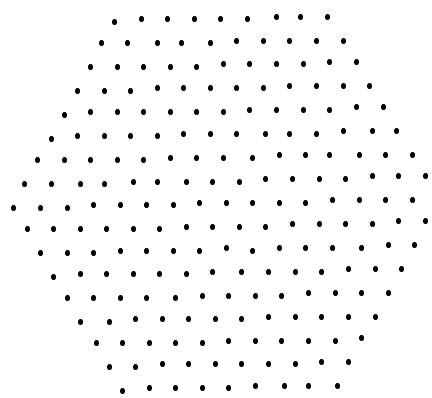
the system was allowed to run for 4000 time steps. Figure 13 shows the final configuration, which is very impressive. The 225 particles have been topologically sorted along the rows and columns of the square lattice. In fact, we have succeeded in creating a perfect square with the 225 particles! AP has performed two tasks simultaneously – forming a square lattice while sorting the particles that form that lattice. The additional task of sorting provides a mechanism for achieving a globally perfect lattice (we confirmed this over ten independent runs).

Since we were motivated initially by the formation of hexagonal structures, we wondered whether the same sorting mechanism could be used to create perfect hexagons. It turns out that it can in fact be done quite simply, if the (m, n) attributes are defined properly. Consider Figure 14, which shows the (m, n) attributes for a simple hexagon composed of seven particles. Once again m and n should not be considered to be coordinates (in fact it is obvious that if they were coordinates they would be incorrect). Rather, they indicate simply the relative ordering of the particles in the 2D plane. The setting of the (m, n) attribute can be easily generalized to hexagons with more particles.

Figure 14. Seven sorted particles.

To test these ideas we chose a system of 217 particles (which is the number of particles needed to create a perfect hexagon with nine particles per side at the outer perimeter) and initialized their (m, n) attribute accordingly. Again $G = 600$ and the system of 217 particles was randomly ini-

tialized. This system was slower to stabilize than the square lattice, and was allowed to run for 15000 time steps. Figure 15 shows the final configuration, which is again very impressive. The 217 particles have been topologically sorted, and have succeeded in forming a perfect hexagon (again we confirmed this over ten independent runs).

7. Summary and Related Work

This paper has introduced a novel framework for distributed control, based on laws of artificial physics (AP). The motivation for this approach is that natural laws of physics satisfy the requirements of distributed control (i.e., self-assembly, fault-tolerance, and self-repair).

The initial results with this framework have been quite promising. We illustrated how a simple AP framework can result in the self-assembly of hexagonal and square lattices. The concept of spin-flipping from natural physics was shown to be useful as a repair mechanism for square lattices, if no global information is available. We have also used these mechanisms to create other structures (e.g., tiling a 2D surface with "open" hexagons that have no particle in the center, by using particles of alternating spin).

The paper has also shown that self-assembly can be viewed as a form of computation, when we use the AP framework to perform sorting. Sorting requires only one small piece of global information – each particle must be able to sense its global orientation. Sorting turned out to be the key to building hexagonal and square lattices that are globally perfect, which brought us back full circle.

Others have examined physical simulations of self-assembly. Schwartz et al. [10] has investigated the self-assembly of viral capsids in a 3D solution, using a kinetics model to simulate the binding of proteins. Winfree [12] has investigated the self-assembly of DNA double-crossover

molecules on a 2D lattice, using a thermodynamic and kinetic model to describe the binding of the molecules. Interestingly, Winfree also shows that self-assembly of DNA is a form of computation, and outlines a simple algorithm that uses self-assembly to solve Hamiltonian Circuit problems. One of the computational steps is a sort, similar to that described in this paper.

Both Schwartz et al. and Winfree are restricted to using plausible models of natural physics, since they are investigating the self-assembly of small natural particles. AP, however, is not bound by this restriction. Since agents have their own sensors and effectors, they can make use of *any* AP force that they can perceive and respond to.

AP is also closely related to the work of Carlson et al. [4], which investigates techniques for controlling miniature agents such as micro-electromechanical agents and nanobots. Their work relies heavily on the use of a global controller that can impose an external potential field that agents can sense. Since we rely primarily on local force interactions, the work by Carlson et al. is complementary.

AP bears some similarity to work in robotics, such as "potential field" and "behavior-based" approaches. Potential field (PF) approaches [6, 7] are used for robot navigation and obstacle avoidance. In a manner similar to AP, PF approaches model a goal position as an attractive force, while obstacles are modeled with repulsive forces. PF computes force vectors by taking the gradient of an entire potential field. In AP, however, each particle directly computes the force vector that applies to its current position – the potential field is never computed. AP thus has lower computational overhead.

Behavior-based approaches (such as motor schema approaches [2] and other ethological behavior-based approaches [3, 8]), derive vector information in a fashion similar to AP. Furthermore, particular behaviors such as "aggregation" and "dispersion" have some similarity to the attractive and repulsive forces in AP. However, behavior-based approaches do not make use of potential fields and forces. Rather, they deal directly with velocity vectors. Although this distinction appears subtle, we believe that it is important for two reasons. First, AP can mimic natural physics phenomena more easily, since it deals directly with forces (e.g., we are not aware of any behavior-based approaches that show clustering or phase transition behavior). Secondly, AP has the potential of being analyzable with conventional physics techniques. In summary, AP potentially places the behavior-based approaches on a firmer physics foundation, yet avoids computing entire potential fields.

The term "artificial physics" has been used in another context, namely, in philosophical discussions concerning the artificial reality necessary to construct artificial life [9]. We use the term more generally to refer to any quasi-natural physics model that we build to solve any particular task.

8. Discussion

One important consequence of these results is the deep connection between distributed control and natural distributed computation, especially if that control is performed using control laws based on AP laws. This observation has two ramifications. First, control systems based on AP laws are likely to display behavior similar to natural systems. This was demonstrated in this paper by illustrating a "phase transition" in the effect that G has on clustering. One could also easily imagine that such control systems will obey various conservation laws, as with natural physical systems. The bottom line is that systems based on AP are likely to exhibit the known characteristics of natural physical systems, which we can use to our advantage.

The second ramification is that it is very likely that the behavior of systems controlled by artificial physics will be amenable to fairly standard analysis tools already used by physicists. Given the difficulty in analyzing complex adaptive systems, many have already taken the approach of using techniques from physics (e.g., statistical mechanics). These techniques are more likely to be appropriate if the systems to which they are applied are similar to the natural physical world. Future work will explore these ramifications.

References

[1] L. Adleman. Computing with DNA. *Sci. Amer.*, Aug, 1998.

[2] T. Balch. *Behavioral diversity in learning robot teams*. PhD thesis, Georgia Inst. of Tech., 1998.

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

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

[5] N. Gershenfeld and I. Chuang. Quantum computing with molecules. *Sci. Amer.*, June, 1998.

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

[7] J. Kim and P. Khosla. Real-time obstacle avoidance using harmonic potential functions. In *IEEE Int'l Conf. on Robotics and Autom.*, pages 790–796, 1991.

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

[9] H. Pattee. Artificial life needs a real epistemology. In Moran, Moreno, Merelo, and Chacon, editors, *Advances in Artificial Life*, pages 23–38. Springer-Verlag, 1995.

[10] R. Schwartz, P. Shor, P. Prevelige, and B. Berger. Local rules simulation of the kinetics of virus capsid self-assembly. *Biophysics*, 75:2626–2636, 1998.

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

[12] E. Winfree. Simulations of computing by self-assembly. *DIMACS: DNA-Based Computers*, June 1998.