Two Formal Gas Models For Multi-Agent Sweeping and Obstacle Avoidance

Wesley Kerr[1], Diana Spears[1], William Spears[1], and David Thayer[2]

[1] Department of Computer Science
[2] Department of Physics and Astronomy
University of Wyoming, Laramie, WY 82071
wkerr@cs.uwyo.edu

Abstract. The task addressed here is a dynamic search through a bounded region, while avoiding multiple large obstacles, such as buildings. In the case of limited sensors and communication, maintaining spatial coverage – especially after passing the obstacles – is a challenging problem. Here, we investigate two physics-based approaches to solving this task with multiple simulated mobile robots, one based on artificial forces and the other based on the kinetic theory of gases. The desired behavior is achieved with both methods, and a comparison is made between them. Because both approaches are physics-based, formal assurances about the multi-robot behavior are straightforward, and are included in the paper.

1 The Sweeping and Obstacle Avoidance Task

The task being addressed is that of sweeping a large group of mobile robots through a long bounded region (a swath of land, a corridor in a building, a city sector, or an underground passageway/tunnel), to perform a search, i.e., surveillance. This requires maximum coverage. The robots (also called "agents") are assumed to lack any active communication capability (e.g., for stealth), and to have a limited sensing range for detecting other agents/objects. It is assumed that robots near the corridor boundaries can detect these boundaries, and that all robots can sense the global direction that they are to move. As they move, the robots need to avoid large obstacles (e.g., buildings). This search might be for enemy mines, survivors of a collapsed building or, alternatively, the robots might be patrolling the area. It is assumed that the robots need to keep moving, because there are not enough of them to view the entire length of the region at once. In other words, the robots begin scattered randomly at one end of the corridor and move to the opposite end (considered the goal direction). This is a "sweep." Once the robots get to the far end of the corridor, they reverse their goal direction and sweep again. Finally, if stealth is an issue then we would like the individual robot movements to be unpredictable to adversaries. It is conjectured that the behavior of a gas is most appropriate for solving this task, i.e., each robot is modeled as a gas particle.

2 Prior Approaches

There are many different methods for controlling groups of autonomous agents (swarms). Balch and Arkin [1] present a very popular approach – using behavior-based techniques. Behavior-based control uses a layered architecture based on arbitration between a suite of behaviors, such as avoidance, exploration, and planning. Although this technique has been successful in maintaining agent formations while going around obstacles, unfortunately it requires a lot of active communication and, typically, it requires small groups of heterogeneous agents that have prespecified roles. Fredslund and Matarić [2] present another behavior based technique using local interactions to create formations and avoid obstacles. This approach has already been ported to robots and experimental results show its successes at avoiding obstacles that are roughly the same size as the robots themselves. However, no solution is presented for the challenging case where the obstacle is the size of a city building.

Other research uses ethological models such as ants or bees to control the robots. In one such study [3], agents are modeled as individual ants in the colony. In this study, the robots leave long-term traces in the environment and require directed graphs to be imposed onto the terrain.

The approaches to swarm control that are of interest to us are rooted in physics. Spears and Gordon [4] have provided a technique called *physicomimetics* for controlling large groups of agents (modeled as particles), using virtual physics-based forces to move the agents into a desired formation, e.g., a hexagonal lattice. This technique scales well to large groups of agents and uses only local interactions. Using physicomimetics, agent swarms do a very nice job of staying in formation and avoiding obstacles, without the need for active communication, long-range sensing, or prespecified roles [5]. Nevertheless, a problem still exists when the agents are presented with a very *large* obstacle, e.g., a building in a city. As the agents move around the obstacle, they are unable to detect the agents that have chosen to move around the other side of the obstacle. Because of this, they are never able to regroup and leave an exposed and uncovered area downstream of the obstacle. The problem is that physicomimetics has traditionally been run in a mode that mimics the behavior of a crystalline solid. Yet solids are rigid and do not expand to fill/cover a region. This is the reason for investigating a gas approach to physicomimetics. The approach of Decuyper and Keymeulen [6] shows that a fluid metaphor works for solving arbitrarily complex mazes. The idea behind this research is that particles in a fluid automatically adapt to changes in the environment because of the fluid's dynamics. The research of Decuyper and Keymeulen has proven that the fluid metaphor is effective, but their approach requires a *global* grid in order to compute the fluid flow through the system. Our research, on the other hand, applies this same fluid metaphor, but using only *local* interactions.

3 Motivation for Gas Models

Both liquids and gases are considered fluids, but this paper focuses on gases. Gases offer excellent coverage, unpredictability of particle locations, and they can be bounded. In general, fluids (gases and liquids) are able to take the shape of their container and therefore are well suited to avoiding obstacles. Fluids are also capable of squeezing through narrow passages and then resuming full coverage when the passage expands. With gases, if we model a container, the gas will eventually diffuse throughout the container until it reaches an asymptotic state. Because gases have this property but liquids do not, gases are a more natural way to think of how to get particles around an obstacle, and why we chose to model a gas. Once the particles have moved around an obstacle, fluids have the ability to regroup. For example, consider releasing a gas from a container at the top of a room with obstacles. The gas inside the container is slightly heavier than the surrounding air. As the gas slowly falls to the ground, it separates around obstacles and expands back to cover areas under the obstacles.

Agents capable of mimicking fluid flow will be successful at avoiding obstacles and moving around them quickly. By mimicking gas flow in particular, the agents will be able to distribute themselves throughout the volume once they have navigated around the obstacle.

This article presents two formal gas models to solve the problem described above, and then compares them. The first approach is physicomimetics, also called *artificial physics (AP)*. The second is *kinetic theory (KT)*, which models virtual inter-particle and particle-wall collisions. Both of these approaches are amenable to straightforward physics analyses for providing behavioral assurances of the robot collective [7], [8].

4 The Physicomimetics Approach

Spears and Gordon [4] have created the artificial physics (AP) framework to control groups of autonomous agents. The goal of AP is one of reducing the potential energy of a system. Each agent in the system experiences a repulsive force from other agents that are too close, and an attractive force from other agents that are too far away. These forces, which are based on Newtonian physics, do not really exist in a physical sense, but the agents react to them as if they were real. Each agent can be described by a position vector $\boldsymbol{x}$ and a velocity vector $\boldsymbol{v}$. Time is maintained with the scalar variable t. The simulation can be run in either 2D or 3D (to model swarms of micro-air vehicles). Agents in the system update their position, $\boldsymbol{x}$, in discrete time steps, Δt. At each time step, each agent updates its velocity, $\boldsymbol{v}$, based on the vector sum (resultant) of all forces exerted on it by the environment, which includes other agents within visibility range, as well as repulsive forces from obstacles and attractive forces from goals. This velocity, $\boldsymbol{v}$, determines $\Delta\boldsymbol{x}$, i.e., the next move of the agent. In particular, at each time step, the position of each particle undergoes a perturbation $\Delta\boldsymbol{x}$. This perturbation depends on the current velocity, i.e., $\Delta\boldsymbol{x} = \boldsymbol{v}\Delta t$. The velocity

of each particle at each time step also changes by $\Delta \boldsymbol{v}$. The change in velocity is controlled by the force on the particle, i.e., $\Delta \boldsymbol{v} = \boldsymbol{F}\Delta t/m$, where m is the mass of that particle and $\boldsymbol{F}$ is the force on that particle. Note that this is the standard, Newtonian $\boldsymbol{F} = m\boldsymbol{a}$ equation.

By setting system parameters in AP, we can mimic solid, liquid, or gas states, as well as phase transitions between these states [7]. Traditionally, AP models a solid. To model a *gas* with AP, all agents experience purely repulsive forces from other agents as well as from obstacles and the side boundaries of the corridor.[3] Although AP was not designed to be an exact model of a gas, we have found that its behavior does a good job of mimicking a gas.

5 The Kinetic Theory Approach

There are two main methods for modeling fluids: the Eulerian approach, which models the fluid from the perspective of a finite volume fixed in space through which the fluid flows (typically the method of computational fluid dynamics), and the Lagrangian approach, in which the frame of reference moves with the fluid volume (typically the kinetic theory approach) [9]. Because we are constructing a model from the perspective of the agents, we choose the latter. Kinetic theory (KT) is typically applied to plasmas or gases, and here we model a gas. This overview of KT borrows heavily from Garcia [10].

When modeling a gas, the number of particles is problematic, i.e., in a gas at standard temperature and pressure there are 2.687×10^{19} particles in a cubic centimeter. A typical solution is to employ a stochastic model that calculates and updates the probabilities of where the particles are and what their velocities are. This is the basis of KT. One advantage of this model is that it enables us to make stochastic predictions, such as the average behavior of the ensemble. The second advantage is that with real robots, we can implement this with probabilistic robot actions, thereby avoiding predictability of the individual agent.

In KT, particles are treated as possessing no potential energy (i.e., an ideal gas), and collisions with other particles are modeled as purely elastic collisions that maintain conservation of momentum. Using some of the formulas for kinetic theory, we can obtain useful properties of the system. If we allow k to be Boltzmann's constant, such that $k = 1.38 \times 10^{-23}$ J/K, m to be the mass of the particle, and T to be the temperature of the system, then we can define the average speed of any given particle (in 3D) as,

$$\langle v \rangle = \int_0^\infty v f(v) dv = \frac{2\sqrt{2}}{\sqrt{\pi}} \sqrt{\frac{kT}{m}}$$

where $f(v)$ is the probability density function for speed.

Another property we can define for KT is the average kinetic energy of the particles:

$$\langle K \rangle = \langle \frac{1}{2} m v^2 \rangle = \frac{3}{2} kT$$

[3] A frictional force is also included in the AP solid model, but is excluded in gas AP.

Using KT, we are able to model different types of fluid flow. For our simulations, we modeled 2D Couette flow. The original code for this one-sided Couette flow is a translation of code from Garcia [10] to the Java programming language. Figure 1 shows a schematic for this one-sided Couette flow, where we have a fluid moving between two walls – one wall moving with velocity v_{wall}, and the other stationary. Because the fluid is a Newtonian fluid and has viscosity, we see a linear velocity profile across the system. Fluid deformation occurs because of the sheer stress τ, and wall velocity is transferred because of molecular friction on the particles that strike the wall. On the other hand, the particles that strike the non-moving wall will transfer some of their velocity to it. This does not cause the wall to move, since in a Couette flow the walls are assumed to have infinite length and therefore infinite mass. We chose a Couette flow so that we can introduce energy into the system and give the particles a direction to move. This effect is similar to AP modeling a goal force.

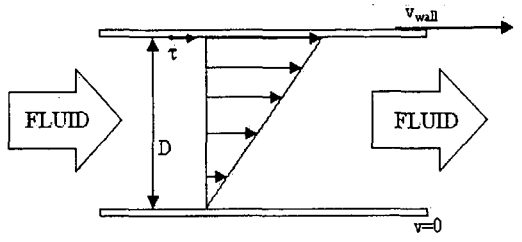

Fig. 1. Schematic for a Couette flow

The main differences between AP and KT are: (1) AP deals with forces. KT deals only with the resulting velocity vectors. (2) With the current force law used by AP, interactions are "soft collisions," i.e., repulsive forces cause small deviations in agent velocities. In KT, collisions cause radical, probabilistic changes in agent velocities. (3) For a given set of starting locations, AP is deterministic, whereas KT is stochastic.

6 Implementation

We created a 2D simulation world with a pair of corridor walls (which can be considered Couette walls), obstacles, and agents (modeled as gas particles). The fluid flow is unsteady with no turbulence, i.e., unsteady laminar flow.

First, we describe our AP gas approach, in which motion is due to attractive and repulsive forces. Recall that AP uses virtual Newtonian force laws. The force law used is:

$$F = |\boldsymbol{F}| = \frac{Gm_1m_2}{r^2} \tag{1}$$

where G is a gravitational constant[4], m_1 and m_2 are the masses, and r is the distance between the agent and another object/agent. For a robotic implementation, there is a maximum possible force, F_{max}, i.e., $F \leq F_{max}$ always. The value of F_{max} used in our simulations is 1.5. The parameter G is set at initialization of the program. To maintain a desired distance, R, between agents in an AP solid, this force is repulsive if $r < R$ and attractive if $r > R$. For an AP gas, the force is always repulsive. Each agent has one sensor to detect the range and bearing to nearby agents, and one effector to move with velocity $\boldsymbol{v}$. To make the simulation a realistic model of robots, agents can only detect other agents/objects within a limited range, namely, $1.5R$. Our implementation assumes $R = 50$.

The corridor and obstacle wall forces are purely repulsive. For AP, the large-scale fluid motion is driven by an attractive goal force at one end of the corridor. Different force constants, G, are allowable for inter-agent forces and agent-wall forces. However, this paper assumes the same G, namely, 1,200. Note that if the forces for avoidance of an obstacle are equal to the attractive forces felt by the goal, the particles reach a stagnation point at the intersection with the obstacle – because all of the forces felt by the particle are in balance. To overcome this situation, when a particle experiences a repulsive force from an obstacle or wall that is the same in magnitude but in the opposite direction of the goal force, the particle translates this into a *tangential* repulsive force. When choosing an angle for the tangential force we must be careful to keep the particle from reaching a stagnation point and keep the particle from moving through the obstacle. Rotating the angle by 45° produces this result nicely. In particular, if the angle of the force is 180° then the angle for this force becomes 135° or 225°, depending on the direction chosen by the robot.

In parallel with the AP approach, we have also implemented the KT approach. Our KT approach models a modified (two-sided) Couette flow in which both Couette walls are moving in the same direction with the same speed. We invented this variant as a means of propelling all agents in a desired general direction, i.e., the large-scale fluid motion becomes that of the walls. Particle velocities start randomly and remain constant, unless collisions occur. (Note that with actual robots, collisions would be virtual, i.e., they would be considered to occur when the agents get too close. Wall motion would also be virtual.) The system updates the world in discrete time steps, Δt. We choose these time steps to occur on the order of the mean collision time for any given agent. Each agent can be described by a position vector $\boldsymbol{x}$ and a velocity vector $\boldsymbol{v}$. At each time step, the position of every agent is reset based on how far it could move in the given time step and its current velocity:

$$\boldsymbol{x} \leftarrow \boldsymbol{x} + \boldsymbol{v}\Delta t \ .$$

This is done for every agent in the system, and positions are updated regardless of walls and obstacles as well as other agents. Once the current agent's position has been updated, a check is performed to see if that agent has moved

[4] G is not related to actual gravity (which is purely attractive), but is a force constant used in the system.

through a wall (including an obstacle wall), in which case the position needs to be reset as if a collision occurred. If the agent strikes a moving wall, then some of the energy from the wall is transferred to the agent. This effect models the molecular friction of the fluid and speeds up the agent. The agent's position is reset as a biased Maxwellian distribution, based on where the agent strikes the wall and how far the agent would have been able to move if the wall were not there. On actual robots, wall collision detection will be done prior to moving. If the robot will intersect with the wall on its next move, then it determines its new position based on a collision, rather than actually colliding with the wall. Once all agents have moved and their positions have been reset based on collisions with the walls, inter-agent collisions are processed. The number of collisions in any given region is a stochastic function of the number of agents in that region (see [10] for details). This process continues indefinitely or until a desired state has been reached.

We have just described the KT approach to modeling Couette flow, modified with a two-sided Couette. We next introduce obstacles into the world, and consider different methods for modeling interactions with obstacle walls.

For one, we could use a KT approach that treats the obstacle boundaries as stationary walls, and processes collisions the same as is done with Couette walls. Unfortunately, in the pure KT approach, agents do not perceive the location of an obstacle until they have collided with it. When colliding with an obstacle, the velocity of the particle off the obstacle is distributed Maxwellian in the goal direction and Gaussian in the lateral direction (i.e., orthogonal to the longitudinal goal direction). This produces excellent results when steady state is reached. A problem arises, however, since we are not modeling a steady state fluid flow. If we were given a steady flow, agents in the system would collide with other agents coming down the flow and through collisions would be pushed around the obstacle. Since flow is unsteady, one of the last agents in the system (i.e., upstream from all the other agents) could strike an obstacle and end up going in the opposite direction with no mechanism to turn it around.

The traditional AP (solid) approach to obstacle avoidance does extremely well at navigating around obstacles. Unfortunately, the AP solid approach does not maintain a good coverage of the environment once the particles have navigated around the obstacle. Figure 2 shows this in simulation. However, the AP *gas* approach (with repulsion only) is able to navigate around obstacles *and* retain good coverage, see Fig. 2. A question remains, nonetheless, as to whether we could do even better by combining AP and KT.

To address this question, we created a hybrid AP/KT algorithm, in which wall collisions generate large-scale motion, AP repulsive forces enable obstacle avoidance, and KT is responsible for agent-agent interactions. By treating the obstacle as a repulsive force, the agents *softly* bounce off the obstacle walls. This force causes the agent to turn, thereby allowing more particles to make it around the obstacle. Since the particles turn softly, they are more likely to hit one of the moving walls and continue in the direction of the flow until they have made it around the obstacle. We are able to achieve an even distribution of particles past

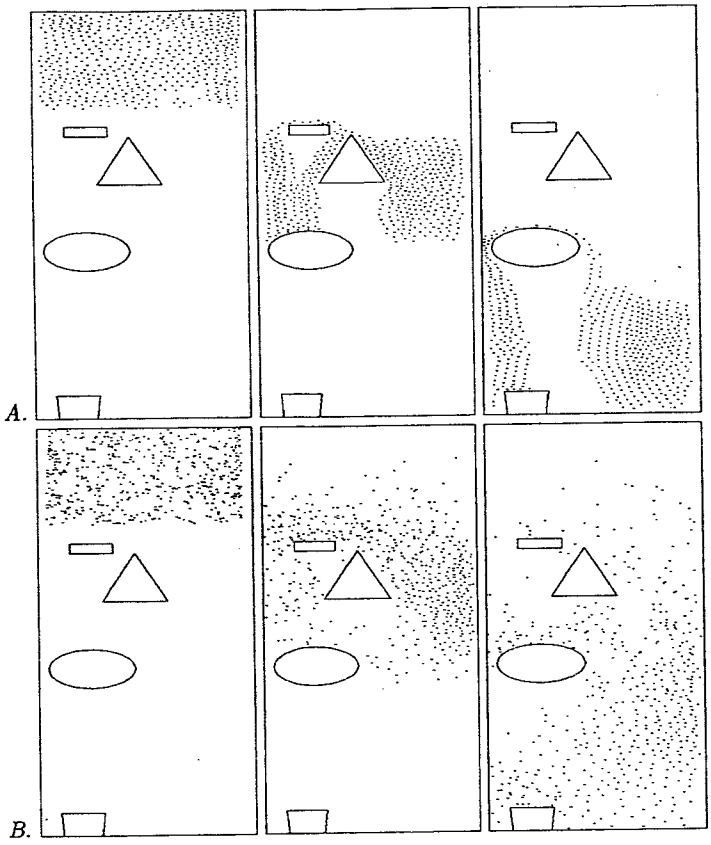

Fig. 2. AP controllers perform a sweep. *A*. AP solid *B*. AP gas

the obstacle with this hybrid, as well as increase the number of particles that make it past in a shorter amount of time. Figure 3 shows the hybrid approach. Note that numerous alternative hybrids of AP and KT are possible; investigation of these others will be a topic for future research.

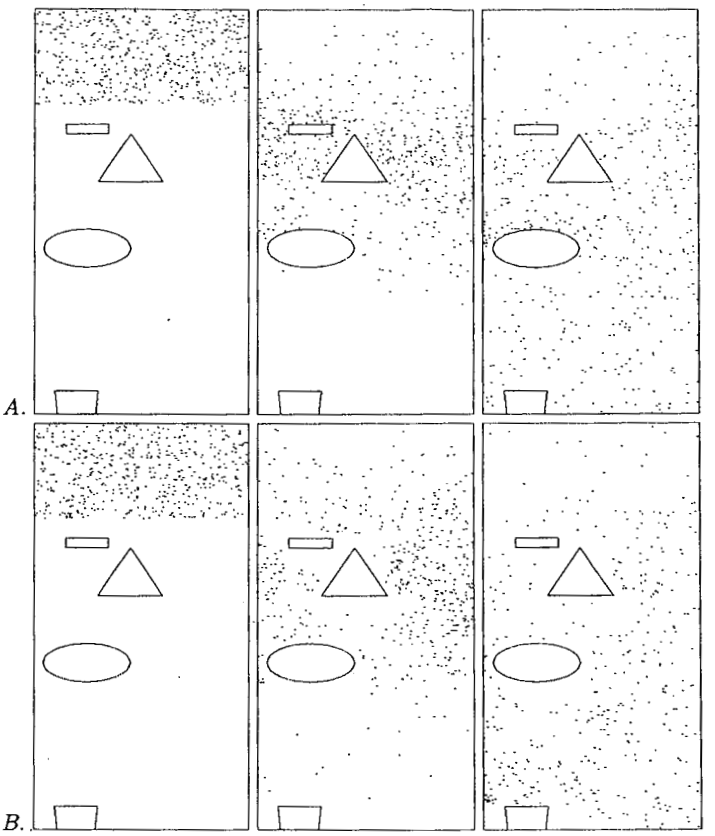

Fig. 3. KT controllers perform a sweep. *A.* KT *B.* AP/KT hybrid

7 Experimental Results

To discover the strengths and weaknesses of each of our four methods (AP solid, AP gas, KT gas, and the AP/KT gas hybrid), we ran numerous empirical experiments with the simulator. Typical results are shown in Figures 2 and 3. In these figure, particles begin at the top and move to the bottom (which is the goal direction). The y-axis is vertical and the x-axis is horizontal. Our starting

point was the AP solid approach to obstacle avoidance. Agent formations stayed intact with this approach, but coverage was very poor. AP gas yielded results far better than AP solid for coverage behind the obstacles (Fig. 2).

Like AP gas, pure KT has yielded excellent coverage. However, problems arose with KT because of the unsteady fluid flow, as discussed above. Furthermore, because of the unsteady nature of the flow, it typically took longer for the entire group of KT particles to get around all of the obstacles (if they were able to do so) than for AP particles to get around the obstacles.

Recall that the hybrid AP/KT approach avoids stagnation points. Other difficulties arise for the AP/KT method. One difficulty arises when two obstacles are very close together, i.e., sufficiently close that the forces exerted from them are able to dominate the goal forces and inter-particle forces. This leaves us with unexplored areas inside our corridor of obstacles (Fig. 4). All methods using force laws had problems dealing with this situation.

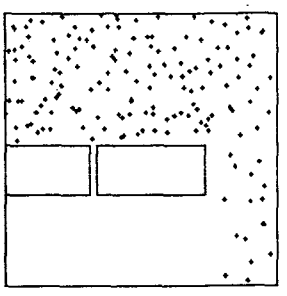

Fig. 4. Obstacle field that has a narrow corridor within. The force-based methods will be unable to explore this area

We have also encountered another potential problem for the KT approaches. The problem does not appear to be due to agent-agent interactions. Rather, the problem arises when trying to address *both* the large-scale movement *and* avoidance of multiple obstacles. We notice this when the obstacle density is increased between the walls. Because the KT methods use collisions with Couette walls for propulsion in a goal direction, the width of the region between these walls determines the coverage of the world. In particular, if the walls contain a group of obstacles several layers abreast, we cannot guarantee that the central region of the Couette, far from the walls, will be covered by the agents. The pure AP models do not have this problem.

In summary, AP solid has very poor coverage, whereas all of the gas models produce excellent coverage, which reaffirms our motivation for choosing gas models. AP and AP/KT hybrid are better than KT for navigating around obstacles, although they have greater difficulty navigating through narrow corridors.

8 Theoretical Predictions

One of the key benefits of using a physics-based multi-agent system is that extensive theoretical (formal) analysis tools already exist for making predictions and guarantees about the behavior of the system. Furthermore, such analyses have the added benefit that their results can be used for setting system parameters for achieving desired multi-agent behavior. The advantages of this are enormous – one can transition directly from theory to a successful robot demo, without all the usual parameter tweaking. For an example of such a success (using AP solid), see [5]. To demonstrate the feasibility of applying physics-based analysis techniques to physics-based systems, we make predictions that support some of our claims regarding the suitability of gas models for our surveillance task.

Before describing the experiments, let us first present the metric used for measuring error between the theoretical predictions and the simulation results. *Relative error* is used, which is defined as:

$$\frac{|\, theoretical - actual \,|}{theoretical}$$

For each experiment, one parameter was perturbed (eight different values of the affected parameter were chosen). For each parameter value, 20 different runs through the simulator were executed, each with different random initial agent positions and velocities. The average relative error (over the 20 runs) and the standard deviation from the average were determined from this sample.

Next, consider the experiments. Recall that our objectives are to sweep a corridor and to avoid obstacles along the way. A third objective for the *swarm* of agents is that of coverage. We define two types of coverage: *longitudinal* (in the goal direction) and *lateral* (orthogonal to the goal direction). Longitudinal coverage can be achieved by movement of the swarm in the goal direction; lateral coverage can be achieved by a uniform spatial distribution of the robots between the side walls. The objective of the surveillance task is to maximize both longitudinal and lateral coverage in the minimum possible time. The number of particles, initial distribution of particles, and termination criterion are determined individually for each experiment, based on earlier studies.

To measure how well the robots achieve the task objective, we observe:

1. The distribution of velocities of all agents in the corridor. This is a measure of both sweep time and total coverage (i.e., a wide distribution typically implies greater coverage of the corridor length and width).
2. The degree to which the spatial distribution of the robots matches a uniform distribution. This is a measure of lateral coverage of the corridor
3. The average agent speed (averaged over all agents in the corridor). This is a measure of total coverage.

Measurement of each of these three aspects of the system (velocity distribution, spatial distribution, average speed) corresponds to each of our three experiments.

Recall (above) that for each experiment, we vary the value of one parameter. The reason for varying such parameter values is to allow a system designer to optimize the design – by understanding the tradeoffs involved. In other words, we have observed that there is a tradeoff between the degrees of longitudinal coverage, lateral coverage, and sweep speed – greater satisfaction of one can lead to reduced satisfaction of the others, making this a Pareto-optimization task. By varying parameter values and showing the resulting velocity and spatial distributions and average speed, a system designer can choose the parameter values that yield desired system performance. Finally, why show *both* theory and simulation results for each experiment and each parameter value? Our rationale is that it is far easier for a system designer to work with the theory when deciding what parameter values to choose for the system. The designer can do this *if* the theory is predictive of the system. In our experimental results below, we show that the theory is indeed predictive of experimental results using our simulation.

For the sake of simplicity, in these experiments we use a subtask of our complete surveillance task. None of the experiments involve obstacles. For the first experiment, the agents are placed uniformly along the beginning of a long corridor and allowed to perform one sweep. In the second experiment, the agents are placed in a square container in an initially tight Gaussian distribution and allowed to diffuse to an asymptotic state. For the final experiment, the agents are placed at the beginning of a long corridor once again, and allowed to run for a predetermined number of time steps, after which the average speed is measured. In the second and third experiments, there is no goal force or wall movement, and therefore there is no directed bulk movement (transport) of the swarm.

8.1 Experiment 1: Velocity Distribution

The first theoretical prediction for our system is devoted to longitudinal coverage and sweep speed via movement. The theory predicts the velocity distribution for each of the approaches, AP and KT. It is assumed that fluid flow is in the y-direction (downward toward the goal), as in Fig. 2.

Recall that the AP approach is an implementation of $\boldsymbol{F} = m\boldsymbol{a}$. Assuming $F_y = g$, where g is the magnitude of the goal force, which is constant for all particles and is strictly in the goal direction, and assuming $m = 1$ (which is assumed throughout this paper), we have the following derivation (where v_y is the magnitude of the velocity in the y-direction, and v_x is assumed to be 0):

$$g = \frac{dv_y}{dt}$$

$$g \cdot dt = dv_y$$

$$g \int dt = \int dv_y$$

$$g \cdot t = v_y$$

$$v_y = gt$$

This shows that the velocity in the direction of the goal is just the force of the goal times the amount of time that has elapsed. We set up an experiment using this theoretical formula to determine the relative error for our experiments. The experiment placed 500 agents in the simulator and terminated in 100 time steps, since by this time the agents reach the maximum velocity that can be achieved on real robots. The parameter being varied is the goal force. The results are plotted in Fig. 5, and the relative error is roughly 1%.

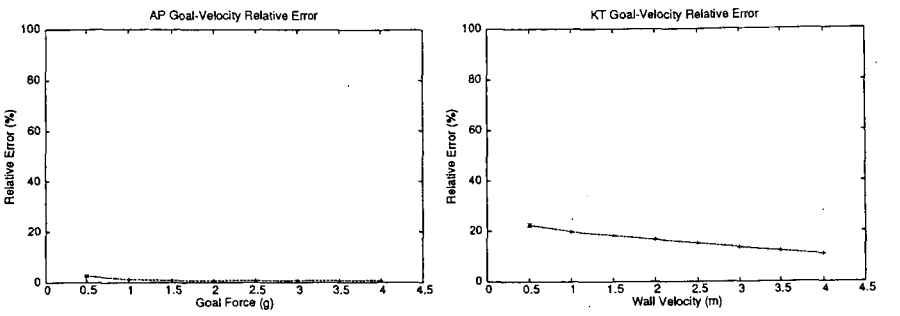

Fig. 5. Relative Error for Goal-Velocity (Prediction 1)

For KT, a traditional one-sided Couette drives the bulk swarm movement. The complete derivation for the velocity profile of a Couette flow can be found in [9] (pages 417–420), but here we present a more concise version.

For steady, 2D flow with no external forces, there is a classical "Governing Equation" that predicts the y-direction momentum of the fluid. This Governing Equation is:

$$\rho v_y \frac{\partial v_y}{\partial y} + \rho v_x \frac{\partial v_y}{\partial x} = -\frac{\partial P}{\partial y} + \frac{\partial \tau_{yy}}{\partial y} + \frac{\partial \tau_{xy}}{\partial x}$$

where ρ is the fluid density, v_x and v_y are the x- and y-components of velocity, P is the fluid pressure, and τ_{yy} and τ_{xy} are the normal and shear stresses, respectively. We can use this equation for momentum to derive the velocity. However, first we need to specialize the equation for our particular situation. For Couette flow, the equation becomes:

$$0 = \frac{\partial}{\partial x}(\mu \frac{\partial v_y}{\partial x})$$

where μ is the fluid viscosity. Assuming an incompressible, constant temperature flow with constant viscosity, this becomes:

$$\frac{\partial^2 v_y}{\partial x^2} = 0 \tag{2}$$

Equation 2 is the Governing Equation for steady, 2D, incompressible, constant temperature Couette flow. Integrating twice with respect to x to find v_y, we get:

$$v_y = c_1 x + c_2 \tag{3}$$

We can solve for c_1 and c_2 from the boundary conditions. In particular, at the stationary Couette wall ($x = 0$), $v_y = 0$, which implies that $c_2 = 0$ from Equation 3. At the moving wall ($x = D$), $v_y = v_{wall}$, where D is the Couette width and v_{wall} is the velocity of the moving wall, which is in the y-direction (toward the goal). Then $c_1 = v_{wall}/D$ from Equation 3.

Substituting these values for c_1 and c_2 back into Equation 3, we get:

$$\frac{v_y}{v_{wall}} = \frac{x}{D}$$

This is a linear profile.

We set up an experiment to measure the relative error generated by our simulation, with each particle behaving as if it were part of a one-sided Couette flow. Each experiment contained 3,000 particles, and ran for 50,000 time steps. When determining the error, we divided the world into seven discrete cells. For each cell, we determined the average velocity of the particles located in that cell. The relative error was averaged across all cells and plotted in Fig. 5 for eight different wall speeds. One can see that the error is below 20%, with a reduction in error for KT as the wall speed is increased. Note that the original algorithms from Garcia [10] also have error between theory and simulation that is slightly below 20%. Reasons for this discrepancy between theory and simulation are elaborated in the discussion section below. When determining the longitudinal coverage via swarm movement, we are able to predict very accurately for both algorithms in the simple scenario, except at slow wall speeds for KT.

8.2 Experiment 2: Spatial Distribution

For the second experiment, we predict the lateral coverage via the spatial distribution. For this experiment, there is neither a goal direction nor obstacles. The agents' task is to diffuse throughout the system. The theory for each approach in gas formation predicts a uniform distribution throughout the system. For the experimental setup, we measured the distance from the uniform distribution once the gas reached an asymptotic state. Therefore, we divided our system into discrete cells and counted the number of particles in each cell. Theory predicts that the number of particles in each cell should be n/c, where n is the total number of particles and c is the total number of grid cells that cover our system.

Our experimental system serves as a simple container to hold a gas. The gas should diffuse within the container until it reaches an asymptotic state and contains equal numbers of particles in each cell. We allowed the system several thousand time steps, starting from a tight Gaussian distribution about the center of the container, to reach this state and then measured the number of particles in each cell. This measurement was averaged over many time steps, since particles

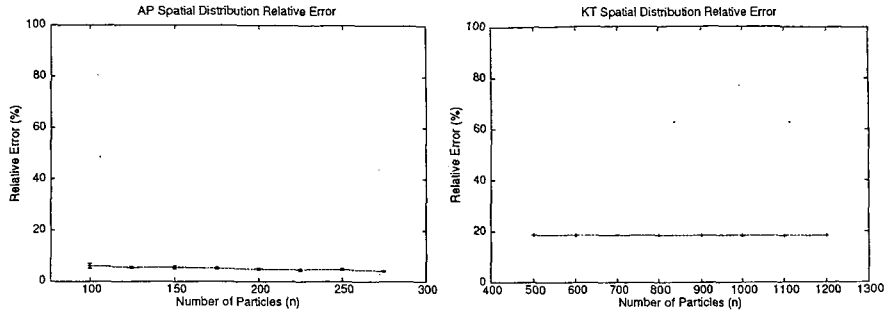

Fig. 6. Relative Error for Uniform Distribution (Prediction 2)

were still moving through the system and diffusion did not imply particles ceased to move. Both experiments were the same for AP and KT, and the results can be found in Fig. 6. In both cases, the parameter being varied is the number of particles. Once again, we are able to predict the spatial distribution with relative error less than 20%.

There is a noticeable downward trend for the relative error in the AP system as more particles are added to the system. Recall that in AP we use forces to affect other particles as well as forces from the walls to keep the particles inside the simulation. This requires that particles have a desired radius such that when another particle enters this radius, it is repelled away. As more particles are added to the simulation, the space is filled with particles that are constantly pushing each other away and moving into the only formation that will allow them all to fit, which is a uniform distribution.

8.3 Experiment 3: Average Speed

For the third experiment, we predict the average speed of the particles in the system. The average speed of the particles serves as a measure of how well the system will be able to achieve complete coverage, because higher speed implies greater coverage. The derivation for AP's prediction of average speed begins with a theoretical formula for AP system potential energy (PE) from [11]. This theory assumes that the particles start in a cluster of radius 0. There are two different situations, depending on the radial extent to which F_{max} dominates the force law $\boldsymbol{F} = m\boldsymbol{a}$. Recall that agents use F_{max} when $F > F_{max}$. This occurs when $\frac{G}{r^2} > F_{max}$ or, equivalently, $r \leq \sqrt{\frac{G}{F_{max}}} \equiv R'$. The first situation is when F_{max} is used only at close distances, i.e., when $0 \leq R' \leq 1.5R$. The second situation occurs when $R' > 1.5R$. Here we assume the first situation, i.e., a low value of G is used such that $G \leq F_{max}(1.5R)^2$, and F_{max} is only used at close distances. Because we are using AP *gas*, there is no friction and all forces are repulsive. We begin with a two-particle system. In this case, the formula is the sum of two integrals. The first represents the force felt by one particle as it

approaches another, from a distance of $1.5R$ to R'. The second is the force F_{max} that is experienced when $0 \leq r \leq R'$. Then, using R' as defined above, with r the inter-agent distance, we have (V is the standard symbol for PE):

$$
\begin{aligned}
PE = V &= \int_0^{R'} F_{max} dr \; + \; \int_{R'}^{1.5R} \frac{G}{r^2} dr \\
&= F_{max} R' \; + \; G \int_{R'}^{1.5R} r^{-2} dr \\
&= F_{max} R' \; + \; G(-r^{-1})|_{R'}^{1.5R} \\
&= F_{max} R' \; + \; G\left(-\frac{1}{1.5R} + \frac{1}{R'}\right) \\
&= \sqrt{GF_{max}} \; + \; G\left(\sqrt{\frac{F_{max}}{G}} - \frac{1}{1.5R}\right) \; because \; R' = \sqrt{\frac{G}{F_{max}}} \\
&= \sqrt{GF_{max}} \; + \; \sqrt{GF_{max}} \; - \; \left(\frac{G}{1.5R}\right) \\
&= 2\sqrt{GF_{max}} \; - \; \left(\frac{G}{1.5R}\right)
\end{aligned}
$$

Now we generalize V to N particles. V_N is our abbreviation for total potential energy, and

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

Note that all the potential energy transforms into kinetic energy (since there is no friction energy dissipation), i.e., $V_N \rightarrow KE$. Also, the total kinetic energy, KE, is equal to $\frac{1}{2}\sum_{i=1}^{N}(v(i))^2$, assuming m=1 and $v(i)$ is the speed of particle i. This formula for KE is equal to $\frac{N}{2}\langle v^2 \rangle$, where $\langle v^2 \rangle$ is the average of the particle speeds squared.
Setting V_N = KE, we get:

$$
\begin{aligned}
\frac{VN(N-1)}{2} &= \frac{N}{2}\langle v^2 \rangle \\
V(N-1) &= \langle v^2 \rangle
\end{aligned}
$$

Substituting for V we get

$$\langle v^2 \rangle \; = \; V(N-1) \; = \; (N-1)\left[2\sqrt{GF_{max}} - \left(\frac{G}{1.5R}\right)\right]$$

From [12], we know that the relationship between $\langle v \rangle$ and $\langle v^2 \rangle$ is the following:

$$\langle v \rangle \; = \; \sqrt{\langle v^2 \rangle - \sigma^2}$$

where σ^2 is the variance of the velocity distribution. However, because the variance of the velocity distribution is not typically available when making a theoretical prediction, one approximation (which is an upper bound on the true theoretical formula because it assumes 0 variance) that we can use is:

$$\langle v \rangle \approx \sqrt{\langle v^2 \rangle} = \sqrt{(N-1)\left[2\sqrt{GF_{max}} - \left(\frac{G}{1.5R}\right)\right]}$$

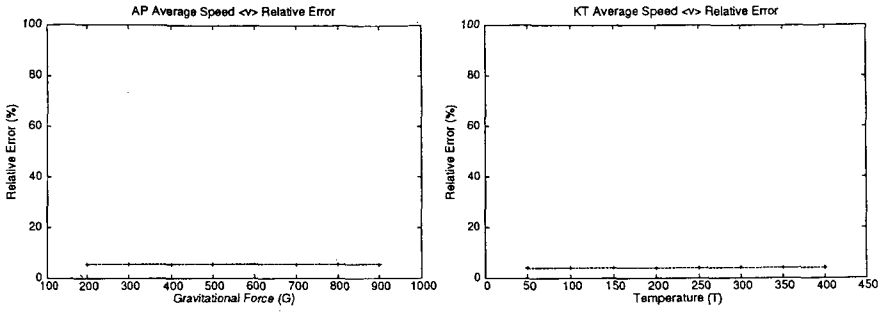

Fig. 7. Relative Error for Average Speed (Prediction 3)

Using this equation for AP, we ran through the experiments (starting with the particles in a tight cluster to match the theory), allowed the gas to reach an asymptotic state, and measured the relative error. For each experiment, there were 100 agents in the system. The total number of time steps required to reach this asymptotic state is different for each value of G since it requires that the agents are no longer interacting with each other. This terminating state can be found when all the agents have ceased to change their velocity. The parameter being varied is the gravitational force, G. As seen in Fig. 7, the error is less than 6%. Furthermore, if the system designer has *any* clue as to what variance to expect in speeds, the theoretical prediction will be greatly improved.

In addition to verifying the formula for $\langle v \rangle$, we also verified the predictiveness of the formula above for $\langle v^2 \rangle$, which is precise because it does not involve variance. The relative error in this case is less than 0.07% for all values of G, which is *extremely* low.

We next show how we derive a KT formula for average speed by modifying the derivation for 3D $\langle v \rangle$ in [10] to a 2D formula for $\langle v \rangle$ (so it applies to our simulation). Assuming a system in thermodynamic equilibrium (since there is no bulk transport), with velocity components within the ranges $v_x + dv_x$ and $v_y + dv_y$, and k is Boltzmann's constant, m is the particle mass, v is the magnitude of the particle velocity (i.e., the particle speed), and T is the initial system temperature (a simple, settable system parameter), then the probabil-

ity, $f(v_x, v_y)dv_x dv_y$, that a particle has velocity components in these ranges is proportional to $e^{(-mv^2/2kT)}dv_x dv_y$. In particular, we have:

$$f(v_x, v_y)dv_x dv_y = Ae^{(-mv^2/2kT)}dv_x dv_y = Ae^{(-m{v_x}^2/2kT)}e^{(-m{v_y}^2/2kT)}dv_x dv_y$$

because $v^2 = {v_x}^2 + {v_y}^2$, and A is a normalization constant that is fixed by the requirement that the integral of the probability over all possible states must be equal to 1, i.e.,

$$\int_0^\infty f(v_x, v_y)dv_x dv_y = 1$$

Therefore,

$$A = \frac{1}{\int_0^\infty e^{(-m{v_x}^2/2kT)}dv_x \int_0^\infty e^{(-m{v_y}^2/2kT)}dv_y}$$

To simplify the expression for A, we can use the fact (from pages 40-46 of [13]) that:

$$\int_0^\infty e^{(-m{v_x}^2/2kT)}dv_x = \sqrt{\frac{2\pi kT}{m}}$$

and then do likewise for v_y. Therefore:

$$f(v_x, v_y)dv_x dv_y = \left(\frac{m}{2\pi kT}\right)(e^{(-m({v_x}^2+{v_y}^2)/2kT)})dv_x dv_y$$

where $\frac{m}{2\pi kT}$ is A.

Note, however, that $f(v_x, v_y)dv_x dv_y$ is a probability for a velocity *vector*, but we want average *speed*. To get average speed, the math is easier if we go from Cartesian to polar coordinates. In particular, to go from velocity to speed, we integrate over all angles.

In polar coordinates, $2\pi v dv$ is the area of extension (annulus) due to Δv. In other words, the area of an annulus whose inner radius is v and outer radius is $v + dv$ is $2\pi v dv$. Then the Maxwell-Boltzmann distribution of speeds, $f(v)dv$, is obtained by integrating the velocity distribution, $f(v_x, v_y)dv_x dv_y$, over all angles from 0 to 2π. This integration yields:

$$f(v)dv = 2\pi v\left(\frac{m}{2\pi kT}\right)(e^{(-mv^2/2kT)})dv$$

Canceling terms, the right-hand side becomes:

$$= v\left(\frac{m}{kT}\right)(e^{(-mv^2/2kT)})dv$$

Because $\langle v \rangle$ is an expected value,

$$\langle v \rangle = \int_0^\infty vf(v)dv = \frac{m}{kT}\int_0^\infty v^2(e^{(-mv^2/2kT)})dv$$

From [14](page 609), we know that $\int_0^\infty e^{-ax^2}x^2dx = \frac{1}{4}\sqrt{\pi}a^{-\frac{3}{2}}$. Substituting v for x and $\frac{m}{2kT}$ for a, we get:

$$\langle v \rangle = \left(\frac{m}{kT}\right) I(2) = \frac{1}{4}\sqrt{\frac{8\pi kT}{m}}$$

Once again, we set up an experiment to measure the actual average speed of the particles in the system. We allowed the system to converge to an asymptotic state for 50,000 time steps measuring the average speed. For each of the 500 particles in the system, we found the average speed, $\langle v \rangle$. This speed was used to find the relative error for the system. Since temperature drives changes in speed, we varied the temperature. Note that by setting T, a system designer can easily achieve desired behavior. The results can be found in Fig. 7 for the different temperatures. Our ability to predict the average speed of the particles is shown, by errors less than 10%.

9 Theoretical Predictions: Discussion

We are capable of predicting three different properties of the system, all of which affect coverage, with an accuracy of less than 20% error, and most with error less than 10%. A 10% error is low for a theoretical prediction.

By looking at the relative error graphs of both the AP and KT approaches, one notices that the AP error is always lower than that of KT (except in the case of $\langle v \rangle$, where the AP formula is a rough approximation). In fact, only KT gets 20% errors – AP errors are always substantially lower than 20%. Our rationale for AP having lower errors between theory and simulation is that AP uses a deterministic agent-positioning algorithm, whereas KT uses a stochastic algorithm for updating particle positions. Therefore, AP predictions are precise, whereas KT predictions are only approximate. Furthermore, as stated in [10], Monte Carlo simulations such as KT need very long runs and huge numbers of particles to acquire enough statistical data to produce accurate (theoretically predictable) results. We cannot guarantee this, since we are developing control algorithms for robotic swarms with a few to a few thousand robots. Therefore, our experiments show a higher error than desired for a Monte Carlo method but they are realistic for real-world swarms.

In conclusion, there appears to be a tradeoff. AP systems are more predictable – both on the macroscopic swarm level and on the level of individual agents. Therefore, if swarm predictability is a higher priority, then AP is preferable. On the other hand, if it is important that individual agents not be predictable (e.g., to an enemy), then KT is preferable.

10 Future Work

The next step is to develop a theory for the full surveillance task. Once this theory is complete, experiments need to be run to test all approaches: AP, KT,

and various AP/KT hybrids. We plan to run numerous experiments to measure coverage versus time and determine which of the algorithms outperforms the others. Once that is complete, the next step is to port these approaches to our laboratory mobile robots. The solid AP approach has already been ported. Transitioning to AP *gas* will be straightforward. We will need to determine, using a more realistic robot simulator, how difficult (or easy) it will be to port KT to the actual robots.

References

1. Balch, T., Arkin, R.: Behavior-based formation control for multi-robot teams. IEEE Trans. on Robotics and Autom. **14** (1998) 1–15
2. Fredslund, J., Matarić, M.: A general algorithm for robot formations using local sensing and mimimal communication. In: IEEE Transactions on Robotics and Automation. (2002) 837–846
3. Koenig, S., Liu, Y.: Terrain coverage with ant robots: A simulation study. In: Agents'01. (2001) 600–607
4. Spears, W., Gordon, D.: Using artificial physics to control agents. In: IEEE International Conference on Information, Intelligence, and Systems. (1999) 281–288
5. Spears, W., Gordon-Spears, D., Hamann, J., Heil, R.: Distributed, physics-based control of swarms of vehicles. Autonomous Robots **17** (2004) 137–162
6. Decuyper, J., Keymeulen, D.: A reactive robot navigation system based on a fluid dynamics metaphor. In Schwefel, H.P., Männer, R., eds.: Parallel Problem Solving from Nature (PPSN I). Volume 496 of Lecture Notes in Computer Science., Springer-Verlag (1991) 356–362
7. Gordon-Spears, D., Spears, W.: Analysis of a phase transition in a physics-based multiagent system. In: Proceedings of FAABS II. (2002)
8. Jantz, S., Doty, K.: Kinetics of robotics: The development of universal metrics in robotic swarms. Technical report, Dept of Electrical Engineering, University of Florida (1997)
9. Anderson, J.: Computational Fluid Dynamics. McGraw-Hill (1995)
10. Garcia, A.: Numerical Methods for Physics. Second edn. Prentice Hall (2000)
11. Spears, W., Spears, D., Heil, R.: A formal analysis of potential energy in a multi-agent system. In: Proceedings of FAABS III. (2004)
12. Stark, H., Woods, J.: Probability, Random Processes, and Estimation Theory for Engineers. Prentice-Hall (1986)
13. Feynman, R., Leighton, R., Sands, M.: The Feynman Lectures on Physics. Addison-Wesley Publishing Company (1963)
14. Reif, F.: Fundamentals of Statistical and Thermal Physics. McGraw-Hill (1965)