International Conference on Control, Automation and Systems 2008
Oct. 14-17, 2008 in COEX, Seoul, Korea

Connectivity of Collaborative Robots in Partially Observable Domains

Suranga Hettiarachchi[1], Paul M. Maxim[2], William M. Spears[2] and Diana F. Spears[2]

[1] Eastern Oregon University, La Grande, Oregon, USA
(E-mail: shettiar@eou.edu)
[2]University of Wyoming, Laramie, Wyoming, USA
(E-mail: paulmax@cs.uwyo.edu, wspears@cs.uwyo.edu, dspears@cs.uwyo.edu)

Abstract: Collaborative information processing is vital for a swarm of robots tasked with many different applications. Swarm connectivity is necessary for achieving good collaboration. Our approach to this is a physics-based autonomous robot framework that acts as a distributed mobile sensor network that is capable of maintaining high connectivity during self-organization and movement. The framework, called *Physicomimetics*, is a robust control scheme built on local interactions between the robots, making it highly scalable, adaptive, and cost effective. This paper presents connectivity results of mobile sensor networks on two partially observable domains – formation movement through obstacle fields and the self-organization of chain formations.

Keywords: collaborative robots, partially observable domains, *Physicomimetics*, chain 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 robot architectures. This combines sensing, computation and networking with mobility, thereby enabling deployment, self-organization, and reconfiguration of the multi-robot collective. Our objective is to provide a scientific, yet practical, approach to the design and analysis of distributed sensor systems.

Numerous tasks require that the swarm of robots maintain high connectivity. One example is the chemical plume tracing (CPT) task [1] that detects a toxic plume and localizes the source emitter. The efficient accomplishment of the CPT task requires that robots maintain their line of sight to other robots in the swarm. The robot collective should be able to maintain high connectivity to reduce the loss of information that is required to succeed at this task. The CPT task mentioned here is an example of a specific problem where robot collaboration improves performance of the system, by acting as a distributed computational sensor network. In this paper we will focus on two general tasks – the movement of a swarm of robots through an obstacle field towards a goal, and the self-organization of chain formations.

2. BACKGROUND AND MOTIVATION

Many sensor networks described in the literature thus far are stationary [3]. Stationary sensor networks work well for applications which involve either long-term monitoring of a specific region (e.g. [4] and others), or detecting unexpected transient events, as in sniper localization [5]. Typically such networks are assumed to consist of a large number of gents, and are optimized to maximize the operating lifetime of the networks, by minimizing the collaborative capabilities of the robots.

In many applications, high connectivity of a sensor network of mobile robots has distinct advantages. The CPT task is one such application. A network of collaborative robots can trace the chemical plume to its source (a goal) while navigating around obstacles in an efficient and robust manner. The mobility and the connectivity of the sensor network allows for fewer robots than in the stationary case, and can adapt to changing environmental conditions.

Our approach to the distributed control of autonomous systems is called *Physicomimetics*. We use the term *Physicomimetics* because although we are motivated by natural physical forces, we are not restricted to them. Although the forces are virtual, robots act as if they were real. Thus, the robot's sensors must see enough to allow it to compute the force to which it is reacting. The robot's effectors must allow it to respond to this perceived force.

There are two potential advantages to the *Physicomimetics* 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. Second, theoretical results translate directly into practical advice on how to set system parameters for desired system performance. This makes the robotic implementation straightforward and its deployment rapid.

This paper is organized as follows. First, we present the general *Physicomimetics* framework. Second, we consider the general task of moving swarm formations through obstacle fields towards a goal. Third, we consider the general task of self-organization of chain formations. Both domains are partially observable. Finally, we present our conclusions.

3. PHYSICOMIMETICS FRAMEWORK

In our *Physicomimetics* framework, virtual physics forces drive the sensor network to a desired configuration or state. The desired configuration is one that minimizes overall system potential energy, and the system acts as a molecular dynamics ($\vec{F} = m\vec{a}$) simulation.

Each robot has position $\vec{p}$ and velocity $\vec{v}$. We use a discrete-time approximation of the continuous behavior of the robots, with time step Δt. At each time step, the position of each robot undergoes a perturbation $\Delta\vec{p}$. The perturbation depends on the current velocity, i.e., $\Delta\vec{p} = \vec{v}\Delta t$. The velocity of each robot at each time step also changes by $\Delta\vec{v}$. The change in velocity is controlled by the force on the robot, i.e., $\Delta\vec{v} = \vec{F}\Delta t/m$, where m is the mass of that robot and $\vec{F}$ is the force on that robot. F and v denote the magnitude of vectors $\vec{F}$ and $\vec{v}$. A frictional force is included, for self-stabilization.

From the start, we wished to have our framework map easily to physical hardware, and our model reflects this design philosophy. Having a mass m associated with each robot 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, in the real robot system. 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 Δt is small, the robots get readings often, whereas if the time step is large, readings are obtained infrequently. We have included a parameter F_{max}, which provides a necessary restriction on the acceleration a robot can achieve. Also a parameter V_{max} restricts the maximum velocity of the robots (and can always be scaled appropriately with Δt to ensure smooth path trajectories).

4. MOVEMENT AROUND OBSTACLES

In prior work we have shown how *Physicomimetics* can be applied to self-organize swarms of robots into hexagonal lattices [6], while they move toward a goal [7], [8] (see Fig. 1). We have also shown that these sensor networks of robots can learn to survive and reach a goal in partially observable domains while avoiding obstacles [9]. In order to accomplish this, robots must be able to sense the range and bearing to nearby robots and obstacles, as well as the goal location. All movement is controlled via the $\vec{F} = m\vec{a}$ control law.

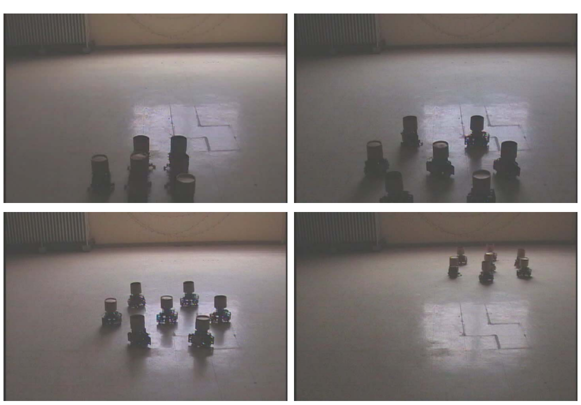

Fig. 1 Seven robots form a hexagon, and move towards a light source.

4.1 Obstructed Perception

When a robot can not sense another robot, due to the presence of obstacles, we call this "obstructed perception." Due to obstructed perception, our robots act in a non-deterministic partially observable domains. Fig. 2 shows an example scenario of obstructed perception. The larger circle C represents an obstacle, and A and B are robots. We define $minD$ to be the minimum distance from the center of the obstacle to the line of sight between robots A and B, and r is the obstacle radius. If $r > minD$, the robots have their perception obstructed.

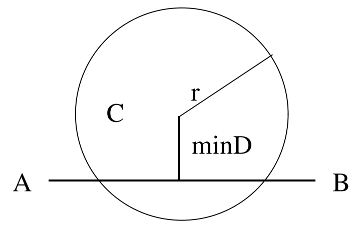

Fig. 2 The sensing capability of two robots A and B is obstructed by a large obstacle C.

We utilize a parameterized description of a line segment [11] to find the $minD$.

$$term_1 = (((1-q) * X_a + q * X_b) - X_c)^2$$
$$term_2 = (((1-q) * Y_a + q * Y_b) - Y_c)^2$$
$$minD = \sqrt{[term_1 + term_2]} \qquad (1)$$

where X_a, X_b are the x positions of robots A and B, Y_a, Y_b are the y positions of robots A and B, X_c and Y_c are the x and y positions of the center of an obstacle, and q is the minimum function that is defined by

$$\frac{((X_c - X_a) * (X_b - X_a) + (Y_c - Y_a) * (Y_b - Y_a))}{\left((X_b - X_a)^2 + (Y_b - Y_a)^2\right)} \qquad (2)$$

4.2 Experimental Set-Up

To achieve the best performance, we optimized the *Physicomimetics* parameters using an EA. We utilized a generalized Lennard-Jones (LJ) force law (which models forces between molecules and atoms):

$$F = 24\epsilon \left[\frac{2dR^{12}}{r^{13}} - \frac{cR^5}{r^7}\right] \qquad (3)$$

where r is the actual distance between robots and R is the desired distance.

EAs are optimization algorithms inspired by natural evolution. We mutate and recombine a population of candidate solutions (individuals) based on their performance in our environment. One of the major reasons for using this population-based stochastic algorithm is that it quickly generates individuals that have robust performance. Every individual in the population is a vector of real-valued *Physicomimetics* parameters. In addition to friction, the evolving parameters of the LJ force law are:

- c - non-negative attractive robot-robot parameter,
- d - non-negative repulsive robot-robot parameter,

- e - strength of the robot-robot interactions,
- F_{max} - maximum force of robot-robot interactions,

and similar 4-tuples for obstacle/goal-robot interactions.

Offspring are generated using one-point crossover with a crossover rate of 60%. Mutation adds/subtracts an amount drawn from a $N(0, \delta)$ Gaussian distribution. Each parameter has a $1/L$ probability of being mutated, where L is the length of the individual. Mutation ensures that parameter values stay within accepted ranges.

Since we are using an EA that minimizes, the performance of an individual is measured as a weighted sum of penalties:

$$w_1 P_{Collision} + w_2 P_{NoCohesion} + w_3 P_{NotReachGoal}$$

The weighted fitness function consists of three components: a penalty for collisions, a penalty for lack of cohesion, and a penalty for robots not reaching the goal. Since there is no safety zone around the obstacles [10], a penalty is added to the score if the robots collide with obstacles. The cohesion penalty is derived from the fact that in a good hexagonal lattice, interior robots should have six local neighbors. A penalty occurs if a robot has more or less neighbors. If no robot reaches the goal within the time limit, a penalty occurs.

4.3 The Environment

The simulation tool consists of a Graphical User Interface (GUI), a training module with an EA, and a performance evaluation module. Whether it is optimizing *Physicomimetics* parameters using the EA or evaluating the performance of the optimal parameter settings, the user can observe the behavior of the robots in the environment using the GUI.

Our 2D training environment is 900 × 700, and contains a goal, obstacles and robots. Up to a maximum of 100 robots and 100 static obstacles with one static goal are placed in the environment. The goal is always placed at a random position in the right side of the world, while the robots are initialized in the bottom left area. The obstacles are randomly distributed throughout the environment, but are kept 50 units away from the initial location of the robots, to give the robots the opportunity to first get into formation. Each circular obstacle has radius R_0 of 10, and the square shaped goal is 20 × 20.

When 100 obstacles are placed in the environment, roughly 5% of the environment is covered by the obstacles (similar to the literature norm [10]). The desired separation, R, between robots is 16, and the maximum velocity V_{max} is 20. Fig. 3 shows 40 robots navigating through randomly positioned obstacles. The larger circles are obstacles and the square to the right is the goal. Robots can sense other robots within the distance of $1.5R$, and can sense the obstacles within the distance of R_0+1 (minimum sensing distance). The goal can be sensed at any distance. A robot's perception of another robot is not obstructed by the obstacles in this environment, (i.e. fully observable domain).

Our 2D performance evaluation environment is 1650 × 950, which is larger than the training environment. In

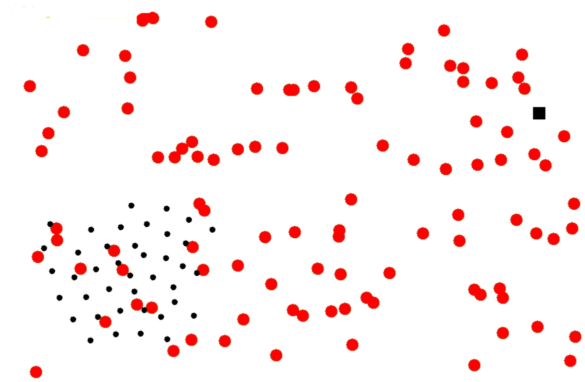

Fig. 3 40 robots moving to the goal. The larger circles represent obstacles, while the square in the upper right represents the goal.

this changed environment, each obstacle has a radius of 30 compared to the obstacle radius of 10 in the training environment. So more than 16% of the performance evaluation environment is covered with the obstacles. Compared to the training environment, the performance evaluation environment triples the obstacle coverage. A robot's perception of another robot is obstructed by the obstacles in this environment (i.e. a partially observable domain), making the task even more difficult.

Robots that are left behind (due to obstacle cul-de-sacs) do not proceed to the goal. We assume that damaged robots can be repaired once they reach the goal.

4.4 Performance Metrics

After optimization, the best force laws are evaluated using our performance module. Our previous work in [12] provides an analysis of swarm connectivity in fully observable domains. In this paper we focus on the connectivity of the swarm in the partially observable domains, where the robots experience much more difficult environments than the ones that they learned during the training.

The swarm connectivity metrics is defined as:

- the maximum number of robots in the swarm that are connected via a communication path. Two robots are connected if their separation is $\leq 1.5R$.

4.5 Results and Analysis

The results presented here focus on the connectivity of the sensor network in partially observable domains. Figs. 4 and 5 illustrate the change in connectivity over 1500 time steps and are averaged over 10 independent runs.

In the first control study , we train 40 robots with an EA on fully observable environments with small obstacles. Fig. 4 shows the connectivity results for 25, 50, 75 and 100 robots in partially observable domains with small obstacles. Interestingly, the robots with partial observability perform well in the environments similar to the ones they learned during training with full observability. This is due to several reasons, 1) the learned environment with small obstacles provides the robots sufficient space to navigate, 2) smaller obstacles cause robot's sensor distraction only by a small fraction of time, and 3) the evolved *Physicomimetics* rules are sufficient for the swarm to navigate around the obstacles to the goal (i.e. similarity of environments).

For all four groups of robots, the connectivity drops after 300 time steps and start increasing after 600 time steps. This is due to the fact that the swarm encounters most of the obstacles between 300 and 600 time steps. This is the middle part of the environment where obstacle density is relatively higher. After 800 time steps, the swarm connectivity is at 100% due to the robots' closeness to the goal.

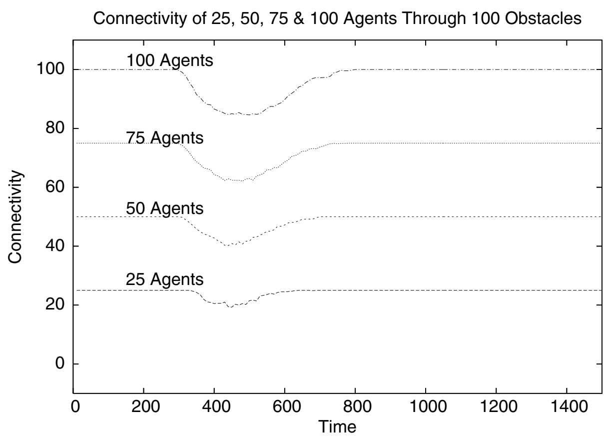

Fig. 4 25, 50, 75 and 100 robots navigating in a partially observable domain towards a goal – small obstacles.

The purpose of the second study is to see how well the knowledge gained while avoiding small obstacles with full observability transferred to large obstacles with partial observability. Fig. 5 shows the connectivity results for 25, 50, 75 and 100 robots in partially observable domains with larger obstacles (three times the prior obstacle density). Again the swarm connectivity reduces around 400 time steps and starts to improve after 900 time steps. The time taken for this improvement is much larger than in the previous study, and the swarm connectivity remains below 100%. The higher obstacle density and obstructed perception cause this reduction in performance. The larger obstacles create large number of cul-de-sacs. These cul-de-sacs and larger obstacles cause much higher sensor obstructions, making navigation much more difficult. Despite these difficulties, however, at least 89% of the original number of robots are connected at the goal. In each case only a small number of robots remain left behind the cul-de-sacs.

In both experiments, despite being trained with only 40 robots, the swarm behavior scales well to larger numbers of robots. This is because the learned behavior of the swarm is to act as a viscous fluid, generally retaining good connectivity while allowing for the deformations necessary to smoothly flow through the obstacle field, despite obstructed perception.

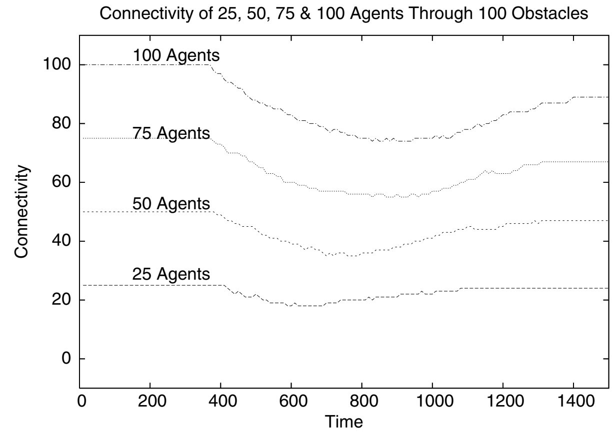

Fig. 5 25, 50, 75 and 100 robots navigating in a partially observable domain towards a goal – larger obstacles.

5. CHAIN FORMATIONS

Another application that requires high connectivity in swarms is "chain formations", where N robots self-organize into a chain to explore environments such as narrow corridors, tunnels and caves. One robot is always kept at the beginning of the environment – that is the first robot in the chain. One of the major challenges in these environments is maintaining a robot's line of sight (i.e., connectivity) to another robot across bends in the environment. Again, we apply our *Physicomimetics* approach to create chain formations of robots in environments. First, a discussion of performance metrics is in order.

5.1 Metrics

We define an optimal chain formation to be one where all robots are connected via communication links, and are extended into the environment as fully as possible. This means that the number of edges are minimized and the length of the edges are maximized. We define this formally as follows.

Each swarm defines a weighted undirected graph G with robots V and communication edges E. The weights are the lengths of the communication links between robots. Let the subgraph G_c represent the robots that are connected via a communication path to the first robot in the chain (including that first robot). This subgraph is represented by the set of robots V_c and their edges E_c. Let $|\cdot|$ represent the cardinality of a set. Also, let G_{MST} represent the minimum spanning tree of G_c. We use four metrics to evaluate the performance of the chain formation algorithm:

- connectivity = $|V_c|$ / N,
- sparseness = $(|V_c| - 1)$ / $|E_c|$,
- coverage = (sum of the edges in G_{MST}) / $R_{max}(N - 1)$,
- fitness = (connectivity + sparseness + coverage)/3.

All metrics range from 0.0 to 1.0, with 1.0 being the optimum (in this situation we use a maximizing EA). The metric we are most interested in is connectivity. This metric provides the fraction of robots that are connected to the start robot via a communication link. However, high connectivity can be achieved with robots that are clustered tightly together – they may not be exploring the environment at all. Because of this we added the two additional metrics sparseness and coverage. First, let us discuss sparseness. In a connected chain, the number of communication links $|E_c|$ should be at most $|V_c| - 1$. If there are more, the chain has redundant links, and the sparseness metric will be less than 1.0. Finally, the chain should explore the environment as much as possible. If the maximum separation between robots is R_{max}, then the maximum distance that can be covered is $R_{max}(N - 1)$. If we sum the edges of the minimum spanning tree of G_c, this provides a reasonable metric of the distance covered by the robots in G_c.

Finally, it is clear that these metrics can be difficult to simultaneously optimize. We provide the fitness metric to indicate the average of connectivity, sparseness, and coverage.

5.2 Modification to Physicomimetics

In most of our implementations of *Physicomimetics*, there is a desired separation R between robots. The sensing range between robots is longer than that. This means that if two robots are closer than R they are repelled from each other. If they are further than R, but are still within sensing range, they are attracted to each other. One can visualize this as two circular disks centered at each robot. One disk has radius R, and the other disk has a larger radius. The inner disk is a zone of repulsion, while the outer annular region (i.e., ring) is a zone of attraction. There is a switch in the sign of the force at R.

One of the guiding principles of *Physicomimetics* is minimalism. An elegant and minimal modification allows for the self-organization of chain formations. To generate chain formations we modified the way the repulsive/attractive switch is performed. We changed the inner circular disk to be an elliptical disk (see Fig. 6).

In this case the force is still attractive within the green region. However, the red portion of the ellipse within the circle is now the repulsive region. The two portions of the ellipse outside the circle are irrelevant, since this is beyond the sensing range. The major axis of the ellipse is aligned with the heading of the robot. The major and minor axis of the ellipse are two of the parameters that we will optimize with an EA.

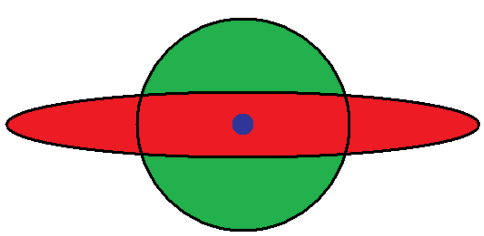

Fig. 6 Elliptical Potential Energy well.

Consider the behavior of two robots that are within sensing range. If they are heading towards each other they will be repelled and move away from each other. If they are not heading towards (or away from) each other they will be attracted. As soon as they are attracted, they will head towards each other, then will be repelled, and will move away. What is interesting is that this also works with large numbers of robots. There will be a period of attractive clustering. Eventually "symmetry breaking" will occur, and a chain formation will emerge.

In the absence of any other sensor information (either global information or local obstacles, for example) the direction of the chain formation will be random. For chain formations in an environment, however, the environment (i.e., the walls of the maze) will bias the formation of the chain towards motion into the maze.

The reader will notice that once the robots repel, they will continue to repel until they are no longer within sensor range. Hence we utilize a "back-flow" force also, which drives a robot towards the location of its closest "upstream" neighbor, if connectivity is lost. One advantage of this approach is that our algorithm also automatically compensates for robots that cease to operate. The chain becomes a bit shorter and connectivity is restored.

5.3 Experimental Set-Up

In order to thoroughly test our chain formation algorithm, we built a simulation tool that allows us to both visualize the robots behavior and also run experiments with various configurations. The graphics can be turned off, allowing for a much shorter execution time. Again, we use an EA to optimize the *Physicomimetics* parameters for chain formations. The force law used is a generalization of the Newtonian force law: $F = G/r^p$.

The EA optimizes the following parameters:

- The friction of the system
- The wall detection range
- The ellipse major and minor axis
- The back flow magnitude and friction
- The Newtonian "robot - robot" G and p parameters
- The Newtonian "robot - wall" G and p parameters
- The front tangential force magnitude

Our simulation models the physical aspects of our Maxelbot robots [13]. Each robot has three Sharp IR sensors for range detection, looking forwards, left, and right. Robot localization is performed via our trilateration module. We model both sensor noise and motor noise, as a multiplicative uniform random variable $[1 - \delta, 1 + \delta]$. For example, 10% noise means that the random variable has range [0.90, 1.10]. Motor noise of 10% means that 10% noise is applied to both the robot turn and the distance the robot travels. Similarly, sensor noise of 10% means that 10% noise is applied to the Sharp IR distance sensor reading, as well as the range and bearing reading from our trilateration module.

Each pair of robots within detection range of each other respond to a "split-Newtonian" force law with parameters p and G ("split" refers to the fact that the force law can be attractive and repulsive). Similarly, robots will respond to walls with a repulsive Newtonian force law with a separate set of p and G parameters. There is no guarantee that a robot can see both sides of a maze corridor. If a robot sees a wall in front, it will apply a tangential force to the left or right (randomly chosen). As mentioned above, in order to promote robustness in the formation, a back flow force was introduced. When a robot loses connectivity with its upstream neighbor, it feels an attractive force towards the last known location of its upstream neighbor.

The simulation world is 750 $\times$ 750 pixels and includes the outline of the environment that is tested, and the robots. The metrics and various parameters are displayed on the screen. We created one rather complex maze for training the parameters with the EA (see Fig. 7). Initially there is only one robot at the beginning of the environment. This robot is stationary, and new robots are added in front of this robot at periodic intervals. As the robots are added they start exploring the environment while maintaining line of sight with their upstream neighbor. The desired distance between the robots is set at $R = 40$, the number of robots is $N = 10$, and the maximum velocity is $V_{max} = 3.0$. Motor and sensor noise were set to 10%. The robots can sense each other within a distance of $1.5R$. Fig. 7 shows a snapshot of the simulation tool with 50 robots.

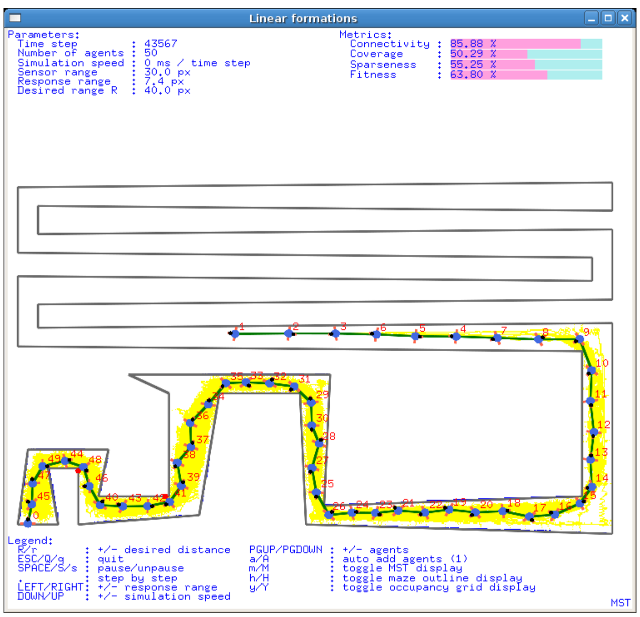

Fig. 7 Simulation tool snapshot.

In order to test the generality of the evolved parameters, eight more environments were built, with varying corridor widths and angle of the bends in the maze. Mazes with branches and loops were also allowed. Two experiments were performed. In the first, we examine the performance as the number of robots N is changed. In the second we changed the amount of sensor and motor noise.

5.4 Experiment I - Variable Number of Robots

Recall that only 10 robots were used during training. To test the scalability of our algorithm, we performed a series of experiments where the number of robots varied from 5 to 50, with an increment of 5. Experiments were run for 100,000 time steps, in order to examine the stability of the chain formations (i.e, to make sure they don't collapse or break apart). Each data point is an average over 10 independent runs. The results of all the four metrics are showed in Figs. 8, 9, 10, and 11.

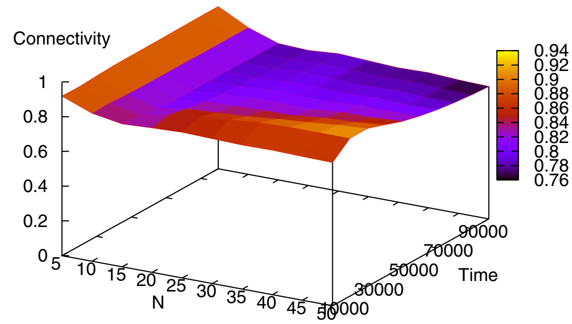

Fig. 8 Connectivity as a function of N and time.

Fig. 8 shows good connectivity results as N increases, indicating that the evolved parameter set is working well with larger numbers of robots. Connectivity is generally better earlier in the simulation, but remains stable after 50,000 time steps.

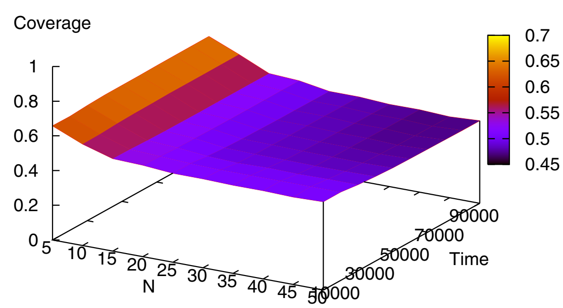

Fig. 9 Coverage as a function of N and time.

Figs. 9 and 10 indicate that good connectivity comes at the expense of somewhat reduced coverage and sparseness, as N increases. This is quite reasonable. As the number of robots increases, the probability of a connectivity link failure in the chain also increases (see mathematical analysis below). To help offset this problem, the resulting chain formations are not stretched as much as possible and possess redundant links.

Fig. 11 indicates that the average of all three metrics is quite acceptable. Average performance falls slowly with increased N and the chain formations are very stable (i.e., time has almost no effect).

5.5 Experiment II - Effect of Motor and Sensor Noise

Our simulation takes in consideration the noise introduced by certain robot hardware components. Noise

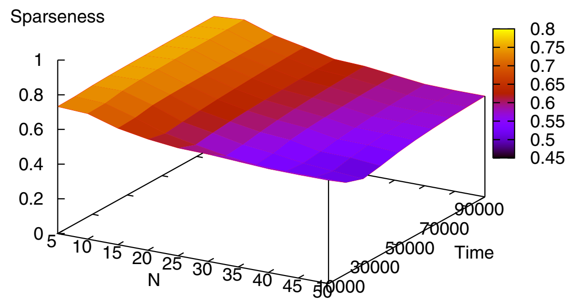

Fig. 10 Sparseness as a function of N and time.

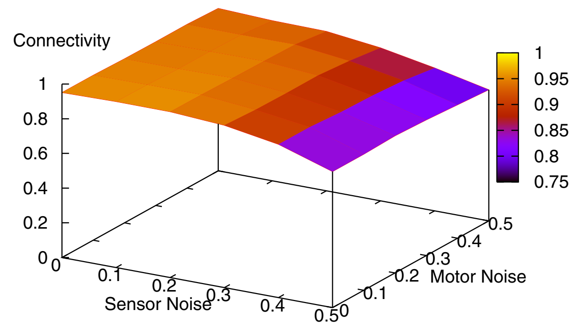

Fig. 12 Connectivity as a function of noise.

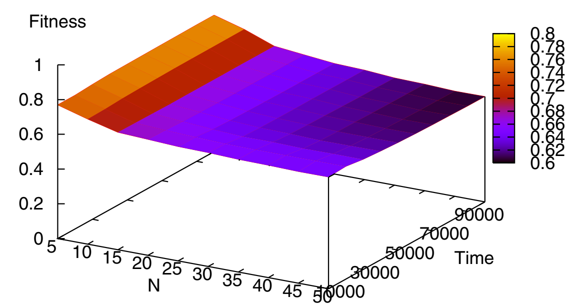

Fig. 11 Fitness as a function of N and time.

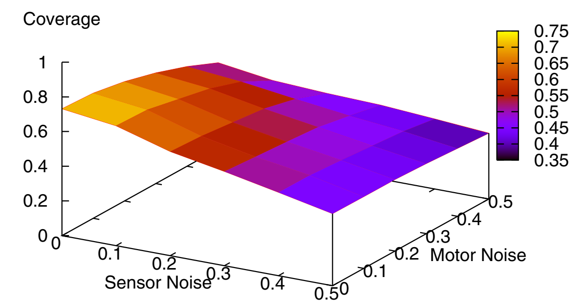

Fig. 13 Coverage as a function of noise.

ranges from 0% to 50%. Sources of noise include the Sharp IR obstacle detection sensors, our trilateration hardware, and the Maxelbot motors. Figs. 12, 13, 14, and 15 illustrate how the performance is affected when noise is introduced in the system. Interestingly, the results paint a similar picture to that shown in the prior experiment.

Once again, connectivity (Fig. 12) is maintained very well, despite the noise. This is achieved because the chain formation does not stretch to its maximum limit (i.e. the robots are compressed to some extent – see Figs. 13 and 14). The overall fitness (Fig. 15) degrades slowly as noise increases, and is least affected by motor noise.

5.6 Analysis of an Ideal Chain Formation

In an ideal chain (that is maximally extended), how hard is it to have chains of length L (on average), given N robots? Let L be the random variable for length, which varies from 0 to $(N-1)$. Let p be the probability that two neighbors are connected. Then by definition, the expected length of the chain is:

$$E[L] = \sum_{l=0}^{(N-1)} l \cdot Prob(l) \tag{4}$$

So,

$$\begin{aligned} E[L] &= 0(1-p) + 1p(1-p) + \cdots + \\ &\quad (N-2)p^{(N-2)}(1-p) + (N-1)p^{(N-1)} \end{aligned}$$

Hence,

$$E[L] = \left[\sum_{l=1}^{N-2} lp^l(1-p)\right] + (N-1)p^{(N-1)} \tag{5}$$

This has an elegant closed form solution,

$$E[L] = \frac{p - p^N}{1-p} \tag{6}$$

As an example with 50 robots, achieving an average chain length of 37 ($\approx 75\%$) requires that each link be "up" 99% of the time.

6. CONCLUSION

Collaborative information processing is vital for a swarm of robots tasked with many different applications. Swarm connectivity is an important aspect for achieving collaboration. Regardless of the formation type, maintaining connectivity among robots in a sensor network is important. In prior work we have shown how our *Physicomimetics* framework can be used to self-organize swarms of mobile robots and generate cohesive sensor networks for fully observable domains.

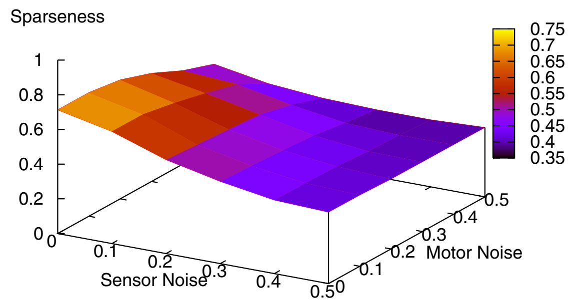

Fig. 14 Sparseness as a function of noise.

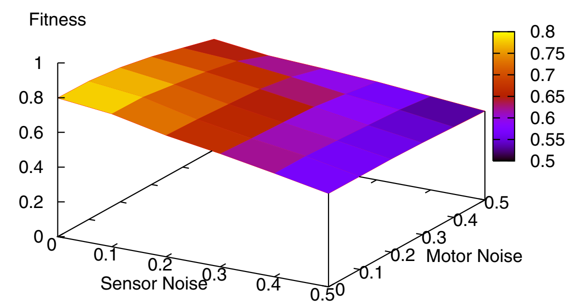

Fig. 15 Fitness as a function of noise.

This paper extends our *Physicomimetics* framework to generate sensor networks with high connectivity in partially observable domains. Two tasks are examined – moving swarm formations through obstacle fields and chain formations. We explain our experimental set-up in detail and summarize how we use evolutionary algorithms to optimize the *Physicomimetics* parameters. In addition, this paper presents several metrics of performance to evaluate the connectivity of sensor network. The results are quite good. The evolved behaviors scale well to increasing numbers of robots as well as increasing domain difficulty.

REFERENCES

[1] Zarzhitsky, D., D. Spears, D. Thayer, and W. Spears. robot-based chemical plume tracing using fluid dynamics. Lecture Notes in Artificial Intelligence, Volume 3228. Springer-Verlag, (2004).

[2] W. Spears, D. Spears, J. Hamann, and R. Heil. Distributed, physicsbased control of swarms of vehicles. In Autonomous Robots, volume 17(2-3), 2004.

[3] I.F.Akyildiz, W.Su, Y.Sankarasubramaniam, E.Cayirci.A survey on sensor networks. IEEE Communications Magazine, Vol.40, Issue 8, pp.102-114, 2002.

[4] A. Mainwaring, J. Polastre, R. Szewcyk, D. Culler, and J. Anderson.Wireless Sensor Networks for Habitat Monitoring, ACM International Workshop on Wireless Sensor Networks and Applications, 2002.

[5] G.Simon. Sensor network-based countersniper system, Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, 2004.

[6] W. Spears and D. Gordon, "Using artificial physics to control robots," in *IEEE International Conference on Information, Intelligence, and Systems*, 1999, pp. 281–288.

[7] W. Spears, R. Heil, D. Spears, and D. Zarzhitsky, "Physicomimetics for mobile robot formations," in *Proceedings of the Third International Joint Conference on Autonomous robots and Multi robot Systems (AAMAS-04)*, 2004, pp. 1528–1529.

[8] W. Spears, D. Spears, J. Hamann, and R. Heil, "Distributed, physics-based control of swarms of vehicles," *Autonomous Robots*, vol. 17, no. 2-3, 2004.

[9] Hettiarachchi S., and W. Spears, DAEDALUS for robots with Obstructed Perception. IEEE Mountain Workshop on Adaptive and Learning Systems, 2006.

[10] T. Balch and M. Hybinette, "Social potentials for scalable multi-robot formations," in *IEEE International Conference on Robotics and Automation*, 2000.

[11] Haeck, N.:Minimum distance between a point and a line, http://www.simdesign.nl/tips/tip001.html. (2002)

[12] Hettiarachchi, S., Spears, W.:Moving swarm formations through obstacle fields. International Conference on Artificial Intelligence. CSREA Press **1** (2005) 97–103

[13] Maxim, P., Hettiarachchi, S., Spears, W., Spears, D., Hamann, J., Kunkel, T., and Speiser, C. Trilateration localization for multi-robot teams. In International Conference on Informatics in Control, Automation, and Robotics, 4th International Workshop on Multi-Agent Robotic Systems (MARS'08). (2008).