Stochastic Swarm Control with Global Inputs

Shiva Shahrokhi and Aaron T. Becker

Abstract— **Simple robots can do huge things. To control swarm of simple robots, we need to know how to steer them simply. One way is global control which means a single input to all the robots (or particles). Previous works show that humans can do it, so we want computers to do it automatically and efficiently.**

I. Introduction

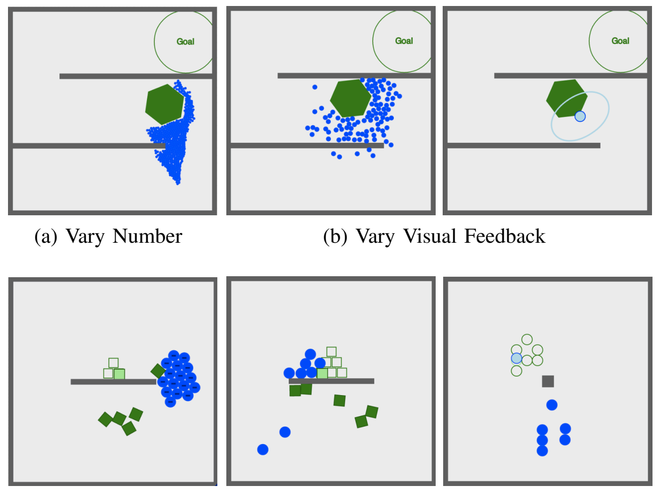
Large populations of micro- and nanorobots are being produced in laboratories around the world, with diverse potential applications in drug delivery and construction [**?**], [**?**], [**?**]. These activities require robots that behave intelligently. Limited computation and communication rules out autonomous operation or direct control over individual units; instead we must rely on global control signals broadcast to the entire robot population. It is not always practical to gather pose information on individual robots for feedback control; the robots might be difficult or impossible to sense individually due to their size and location. However, it is often possible to sense global properties of the group, such as mean position and density. Finally, many promising applications will require direct human control, but user interfaces to thousands—or millions—of robots is a daunting human-swarm interaction (HSI) challenge.

The goal of this work is to provide a tool for investigating HSI methods through statistically significant numbers of experiments. There is currently no comprehensive understanding of user interfaces for controlling multi-robot systems with massive populations. We are particularly motivated by the sharp constraints in micro- and nanorobotic systems. For example, full-state feedback with 10^6 robots leads to operator overload. Similarly, the user interaction required to individually control each robot scales linearly with robot population. Instead, user interaction is often constrained to modifying a global input. This input may be nonstandard, such as the attraction/repulsion field from a scanning tunneling microscope (STM) tip.

Our previous work with over a hundred hardware robots and thousands of simulated robots [**?**] demonstrated that direct human control of large swarms is possible. Unfortunately, the logistical challenges of repeated experiments with over one hundred robots prevented large-scale tests.

Our goal was to test several scenarios involving large-scale human-swarm interaction (HSI), and to do so with a statistically-significant sample size. Towards this end, we created SwarmControl.net, an open-source online testing platform suitable for inexpensive deployment and data collection on a scale not yet seen in swarm robotics research. Screenshots from this platform are shown in Fig. 1. All code [**?**], and experimental results are posted online.

S. Shahrokhi and A. Becker are with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77204-4005 USA {sshahrokhi2, atbecker}@uh.edu



(a) Vary Number (b) Vary Visual Feedback

(c) Vary Control (d) Vary Noise (e) Control Position

Fig. 1. Screenshots from our five online experiments controlling multi-robot systems with limited, global control. **(a)** Varying the number of robots from 1-500 **(b)** Comparing 4 levels of visual feedback **(c)** Comparing 3 control architectures **(d)** Varying noise from 0 to 200% of control authority **(e)** Controlling the position of 1 to 10 robots. See video overview at http://youtu.be/HgNENj3hvEg.

Our experiments show that numerous simple robots responding to global control inputs are directly controllable by a human operator without special training, that the visual feedback of the swarm state should be very simple in order to increase task performance, and that humans perform swarm-object manipulation faster using attractive control schemes than repulsive control schemes.

Our paper is organized as follows. After a discussion of related work in Section II, we describe our experimental methods for an online human-user experiment in Section **??**. We report the results of our experiments in Section V, discuss the lessons learned in Section **??**, and end with concluding remarks in Section VI.

II. Related Work

A. Human-Swarm Interaction

Olson and Wood studied human *fanout*, the number of robots a single human user could control [**?**]. They postulated that the optimal number of robots was approximately the autonomous time divided by the interaction time required by each robot. Their sample problem involved a multi-robot search task, where users could assign goals to robots.

Their user interaction studies with simulated planar robots indicated a *fanout plateau* of about 8 robots, after which there were diminishing returns. They hypothesize that the location of this plateau is highly dependent on the underlying task, and our work indicated there are some tasks without plateaus. Their research investigated robots with 3 levels of autonomy. We use robots without autonomy, corresponding with their first-level robots.

Squire, Trafton, and Parasuraman designed experiments showing that user-interface design had a high impact on the task effectiveness and the number of robots that could be controlled simultaneously in a multi-robot task [**?**].

A number of user studies compare methods for controlling large swarms of simulated robots, for example [**?**], [**?**], [**?**]. These studies provide insights but are limited by cost to small user studies; have a closed-source code base; and focus on controlling intelligent, programmable agents. For instance [**?**] was limited to a pool of 18 participants, [**?**] 5, and [**?**] 32. Using an online testing environment, we conduct similar studies but with much larger sample sizes.

B. Global-control of micro- and nanorobots

Small robots have been constructed with physical heterogeneity so that they respond differently to a global, broadcast control signal. Examples include *scratch-drive microrobots*, actuated and controlled by a DC voltage signal from a substrate [**?**], [**?**]; magnetic structures with different cross-sections that could be independently steered [**?**], [**?**]; *MagMite* microrobots with different resonant frequencies and a global magnetic field [**?**]; and magnetically controlled nanoscale helical screws constructed to stop movement at different cutoff frequencies of a global magnetic field [**?**], [**?**].

Similarly, our previous work [**?**], [**?**] focused on exploiting inhomogeneity between robots. These control algorithms theoretically apply to any number of robots—even robotic continuums—but in practice process noise cancels the differentiating effects of inhomogeneity for more than tens of robots. We desire control algorithms that extend to many thousands of robots.

C. Three challenges for massive manipulation

While it is now possible to create many micro- and nanorobots, there remain challenges in control, sensing, and computation.

1) Control—global inputs: Many micro- and nanorobotic systems [**?**], [**?**], [**?**], [**?**], [**?**], [**?**], [**?**], [**?**], [**?**], [**?**] rely on global inputs, where each robot receives an exact copy of the control signal. Our experiments follow this global model.

2) Sensing—large populations: Parallel control of n differential-drive robots in a plane requires $3n$ state variables. Even holonomic robots require $2n$ state variables. Numerous methods exist for measuring this state in micro- and nanorobotics. These solutions use computer vision systems to sense position and heading angle, with corresponding challenges of handling missed detections and image registration between detections and robots. These challenges are increased at the nanoscale where sensing competes with control for communication bandwidth. We examine control when the operator has access to partial feedback, including only the first and/or second moments of a population's position, or only the convex-hull containing the robots.

3) Computation—calculating the control law: In our previous work the controllers required at best a summation over all the robot states [**?**] and at worst a matrix inversion [**?**]. These operations become intractable for large populations of robots. By focusing on *human* control of large robot populations, we accentuate computational difficulties because the controllers are implemented by the unaided human operator.

III. Theory

A. Models

We consider holonomic robots that move in the 2D plane. We want to control position and velocity of the robots and deciding about force for reaching our desired position and velocity. So our input is going to be acceleration (*F* = *ma*). If we show acceleration by *a*, velocity by *v* and position in x coordinate by P_x and in y coordinate by P_y, we have the following equations

$$\begin{bmatrix} \dot{P}_x = v_x \\ \dot{v}_x = a_x \end{bmatrix} \tag{1}$$

$$\begin{bmatrix} \dot{P}_y = v_y \\ \dot{v}_y = a_y \end{bmatrix} \tag{2}$$

The state-space representation of our OpenLoop controller is:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y &= Cx(t) + Du(t) \end{aligned} \tag{3}$$

where *x(t)* represents our states, *u(t)* is our input and *e(t)* represents noise in the system. We also have *y* as our output. First we assume that we don't have noise in the system and we also have just one robot. As mentioned, *a* is our input and *x, y,* and their velocities are our states that we want to control.
We define our states as following:

$$\begin{aligned} x_1 &= P_x \\ \dot{x}_1 &= x_2 = \dot{P}_x = v_x \\ x_3 &= P_y \\ \dot{x}_3 &= x_4 = \dot{P}_y = v_y \end{aligned} \tag{4}$$

So our state space representation is:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} u \tag{5}$$

We want to find number of states that we can control. We need to know rank of the controllability matrix $\{B, AB, A^2B, ..., A^{n-1}B\}$.

$$C = \{B, AB, A^2B, ..., A^{n-1}B\} \tag{6}$$

$$C = \left\{ \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \right\} \tag{7}$$

So we showed that we have two controllable states.

B. Independent control is not possible

We know that we can control velocity of our robot, we want to see what happens if we had more than one robot? As we saw here, v_x is completely independent of v_y, so if we wanted to have n robots, we had n states in x axis, and n states in y axis, which were completely independent from each other. So assume we have n robots and want to control them in *x* axis:

$$\begin{aligned} \dot{P}_{x1} &= v_{x1} \\ \dot{v}_{x1} &= a_{x1} \\ \dot{P}_{x2} &= v_{x2} \\ \dot{v}_{x2} &= a_{x2} \\ &\vdots \\ \dot{P}_{xn} &= v_{xn} \\ \dot{v}_{xn} &= a_{xn} \end{aligned} \tag{8}$$

So our state-space representation will be:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ . \\ . \\ . \\ \dot{x}_{2n-1} \\ \dot{x}_{2n} \end{bmatrix} = \begin{bmatrix} 0 & 1 & . & . & . & 0 & 0 \\ 0 & 0 & . & . & . & 0 & 0 \\ 0 & 0 & 0 & 1 & . & 0 & 0 \\ 0 & 0 & 0 & 0 & . & 0 & 0 \\ . & . & . & . & . & . & . \\ 0 & 0 & . & . & . & 0 & 1 \\ 0 & 0 & . & . & . & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_{2n-1} \\ x_{2n} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ . \\ . \\ . \\ 0 \\ 1 \end{bmatrix} a_x \tag{9}$$

If we had n robots, we had exactly the same symmetry of what we had for 1 robot. We can again control two states because we have rank two in C:

$$C = \left\{ \begin{bmatrix} 0 \\ 1 \\ . \\ . \\ . \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ . \\ . \\ . \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ . \\ . \\ . \\ 0 \\ 0 \end{bmatrix}, \ldots \right\} \tag{10}$$

C. Controlling Mean Position

So for any number of robots if we give a global command to them, we have just two controllable states in each axis. So it is obvious that we can not control position of all the robots, but what states are controllable? To answer this question we create a reduced order system that calculates average position and average velocity of the robots:

$$\begin{bmatrix} \dot{\bar{x}}_p \\ \dot{\bar{x}}_v \end{bmatrix} = \frac{1}{n} \begin{bmatrix} 0 & 1 & 0 & 1 & \ldots & 0 & 1 \\ 0 & 0 & 0 & 0 & \ldots & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_{2n-1} \\ x_{2n} \end{bmatrix} + \frac{1}{n} \begin{bmatrix} 0 & 0 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & 1 & \ldots & 0 \end{bmatrix} \tag{11}$$

Thus:

$$\begin{bmatrix} \dot{\bar{x}}_p \\ \dot{\bar{x}}_v \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{x}_p \\ \bar{x}_v \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \tag{12}$$

We analyze *C* for *y*:

$$C = \left\{ \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right\} \tag{13}$$

This matrix again has rank two, and thus all the states are controllable. These controllable states are the average position and average velocity:

$$a = K \left[\begin{bmatrix} \dot{\bar{x}}_p \\ \dot{\bar{x}}_v \end{bmatrix} - \begin{bmatrix} x_{goal} \\ \dot{x}_{goal} \end{bmatrix} \right] \tag{14}$$

■

D. Controlling the variance of many robots

As shown in section ?? above, only the mean position of a group of robots is controllable. However, there are several techniques for breaking symmetry, for example by allowing independent noise sources **??**, or by using obstacles **??**.

Control the variance requires being able to increase and decrease the variance. Given a large free workspace, Brownian noise is sufficient to increase the variance. A flat obstacle can be used to decrease variance.

Control Lyapunov function on $\sigma(t), \sigma_{goal}$, provide a control law.

Real systems, especially at the micro scale, are affected by unmodelled dynamics much of which can be designed by Brownian noise. To model this equation (3) must be modified as follows:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + We(t) \\ y &= Cx(t) + Du(t) \end{aligned} \tag{15}$$

where $e(t)$ is the error in the system.

After some time, gaussian distribution shapes the outline of the robots because of the Brownian noise feature:

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \tag{16}$$

where σ is standard deviation:

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (x_i - \overline{x})^2} \tag{17}$$

After a while, if we have lots of robots, we see the robots to make a bell shaped curve because of the Brownian noise. In we see the probability density function of a Gaussian distribution:

E. Controlling both the mean and the variance of many robots

hysteresis control law, [?]

(18)

IV. Simulation

A. Controlling the mean position

give PID control law, explain experiment (number of robots, maximum speed,).

contrast controllers – as is typical with PID control laws, we can tune the response to meet desired specifications.

image showing varying P control

image showing varying D control

image showing varying the number of robots n

B. Controlling the variance

cite the control law, explain experiment (number of robots, maximum speed,).

contrast controllers – as is typical with PID control laws, we can tune the response to meet desired specifications.

image showing varying Brownian noise

image showing control x variance and y-variance out of phase

C. Hysteresis Control of mean and variance

plot showing 1.5 cycles of mean position, and a variance goal. We might need a longer time

V. Results

The experimental section compares the results of our hysteresis based controller applied to a *block-pushing* task. In preliminary work over 1000 human users completed this task using varying levels of feedback. To our surprise, users who received the lowest amount of feedback – just the moments of the position distribution of the robot swarm – performed better than users with full state feedback.

A. Human-user experiment

Sensing is expensive, especially on the nanoscale. To see nanocars [?], scientists fasten molecules that fluoresce light when activated by a strong light source. Unfortunately, multiple exposures can destroy these molecules, a process called *photobleaching*. Photobleaching can be minimized by lowering the excitation light intensity, but this increases the probability of missed detections [?]. This experiment explores manipulation with varying amounts of sensing information: **full-state** sensing provides the most information by showing the position of all robots; **convex-hull** draws a convex hull around the outermost robots; **mean** provides the average position of the population; and **mean + variance** adds a confidence ellipse. Fig. 3 shows screenshots of the same robot swarm with each type of visual feedback. Full-state requires $2n$ data points for n robots. Convex-hull requires at worst $2n$, but usually a smaller number. Mean requires two, and variance three, data points. Mean and mean + variance are convenient even with millions of robots. Our hypothesis predicted a steady decay in performance as the amount of visual feedback decreased.

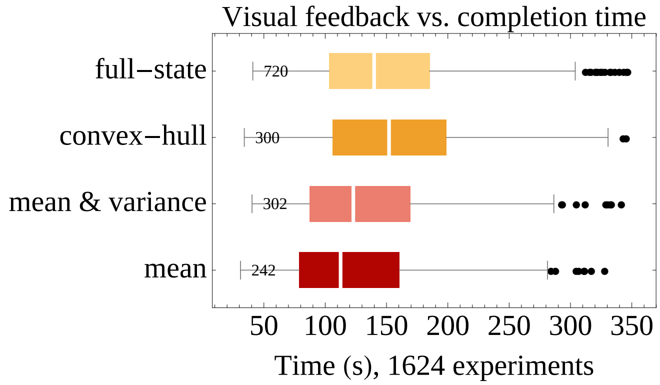


Fig. 2. Completion-time results for the four levels of visual feedback shown in Fig. 3. Surprisingly, players perform better with limited feedback–subjects with only the mean + variance outperformed all others.

To our surprise, our experiment indicates the opposite: players with just the mean completed the task faster than those with full-state feedback. As Fig. 2 shows, the levels of feedback arranged by increasing completion time are [mean + variance, mean, full-state, convex-hull]. Anecdotal evidence from beta-testers who played the game suggests that tracking 100 robots is overwhelming—similar to schooling phenomenons that confuse predators—while working with just the mean + variance is like using a "spongy" manipulator. Our beta-testers found convex-hull feedback confusing and irritating. A single robot left behind an obstacle will stretch the entire hull, obscuring the majority of the swarm.

B. Automated Block Pushing

To solve this problem, the discretized the environment, used breadth-first search to determine M, the shortest path from any point for the block to the goal, and generate a gradient map ∇M toward the goal. The blocks's center of mass is at b and has radius r_b. The robots were then directed to assemble at $b - r_b \nabla M$ to push the block toward the goal location.

TODO: write this in algorithmic form

image: show the vector field and the robots pushing a block

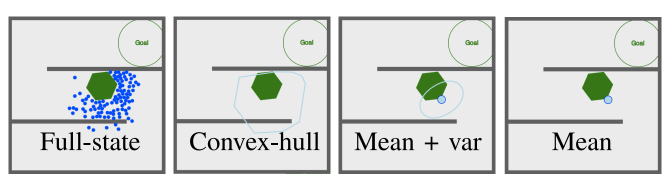


Fig. 3. Screenshots from task *Vary Visualization*. This experiment challenges players to quickly steer 100 robots (blue discs) to push an object (green hexagon) into a goal region. We record the completion time and other statistics.

plot of results, comparing to humans

plot of results, varying the number of robots

images of worlds where the algorithm fails, short discussion.

VI. Conclusion and Future Work

We ...

VII. Acknowledgements

We acknowledge many mrsl.rice.edu lab members who helped beta test SwarmControl.net; helpful discussion with James Tour, Stephan Link, and Victor Garcìa Lòpez on fabrication and visualization challenges at the nanoscale with nanocars; and the Rice Center for Technology in Teaching and Learning. This work was partially supported by the National Science Foundation under CPS-1035716.