

Steering a Swarm of Particles Using Global Inputs and Swarm Statistics

Shiva Shahrokhi, *Student Member, IEEE*, Lillian Lin, *Student Member, IEEE*, Chris Ertel, Mable Wan, and Aaron T. Becker, *Senior Member, IEEE*

Abstract—Microrobotics has the potential to revolutionize many applications including targeted material delivery, assembly, and surgery. The same properties that promise breakthrough solutions—small size and large populations—present unique challenges for controlling motion. Robotic manipulation usually assumes intelligent agents, not particle systems manipulated by a global signal. To identify the key parameters for particle manipulation, we used a collection of online games in which players steer swarms of up to 500 particles to complete manipulation challenges. We recorded statistics from more than 10 000 players. Inspired by techniques in which human operators performed well, we investigate controllers that use only the mean and variance of the swarm. We prove that mean position is controllable and provide conditions under which variance is controllable. We next derive automatic controllers for these and a hysteresis-based switching control to regulate the first two moments of the particle distribution. Finally, we employ these controllers as primitives for an object manipulation task and implement all controllers on 100 kilobots controlled by the direction of a global light source.

Index Terms—Human-swarm interaction, manipulation planning, swarm, underactuated robots.

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, see [1]–[3]. These activities require robots that behave intelligently. Limited computation and communication at small scales makes autonomous operation or direct control over individual robots difficult. Instead, this paper treats the robots as particles that are steered by a global control signal broadcast to the entire population. This paper examines object manipulation by a swarm of particles, as illustrated in Fig. 1. The transportation methodology is similar to that in [4], but rather than using onboard

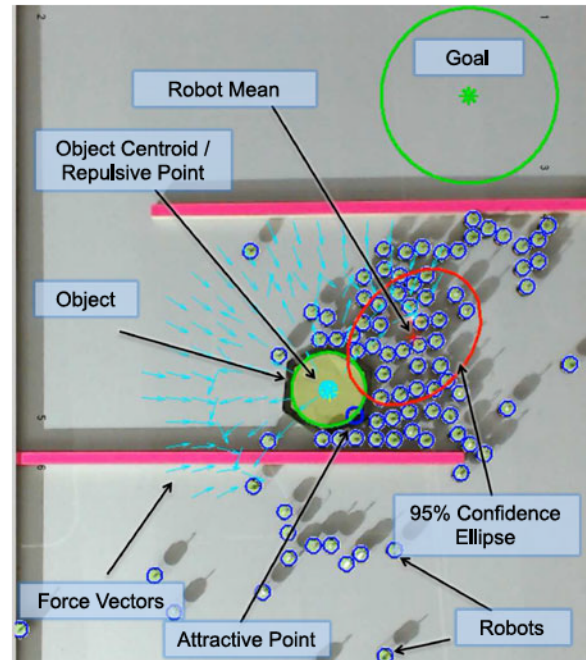


Fig. 1. Swarm of particles, all actuated by a uniform control input where each particle gets the same control input, can be effectively manipulated by a control law that uses only the mean and variance of the robot distribution. Here a swarm of particles (kilobot robots) pushes a green hexagon toward the goal (see video attachment).

computation or sensing, the particles all move in the same direction.

Many promising applications for particle swarms require direct human control, but user interfaces to thousands—or millions—of particles is a daunting human–swarm interaction (HSI) challenge. Our early work with more than a hundred hardware robots and thousands of simulated particles demonstrated that direct human control of large swarms is possible [5]. Unfortunately, the logistical challenges of repeated experiments with more than 100 robots prevented large-scale tests. There is currently no comprehensive understanding of user interfaces for controlling multirobot systems with massive populations. One contribution of this paper is a tool for investigating HSI methods through statistically significant numbers of experiments.

Often particles are difficult or impossible to sense individually due to their size and location. For example, microrobots are smaller than the minimum resolution of a clinical MRI scanner, see [6], however it is often possible to sense global

Manuscript received May 16, 2017; revised October 3, 2017; accepted October 25, 2017. This paper was recommended for publication by Associate Editor L. Pallottino and Editor T. Murphey upon evaluation of the reviewers' comments. This work was supported by the National Science Foundation under Grant IIS-1553063. (Corresponding author: Aaron T. Becker.)

The authors are with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77204 USA (e-mail: sshahrokhi2@uh.edu; lillianlin@ieee.org; ertel.chris@gmail.com; mablewan.mw@gmail.com; atbecker@uh.edu).

This paper has supplementary downloadable material available at <http://ieeexplore.ieee.org>.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TRO.2017.2769094

properties of the group such as mean position and variance. To make progress in automatic control with global inputs, this paper presents swarm manipulation controllers inspired by our online experiments that require only mean and variance measurements of the particle's positions. To perform the object manipulation task illustrated in Fig. 1, we use these controllers as primitives, policy iteration for path planning, handle outliers by partitioning the workspace, and minimize pushing the object backwards with potential field navigation.

This 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 and their results in Section III. Next, we prove that the mean and variance of a particle swarm are controllable in Section IV, and present automatic controllers in Section V. We use these controllers as primitives and present a framework for manipulating an object through a maze in Section VI. We implement these controllers in our hardware robots and use them to complete an object manipulation task with 100 kilobots in Section VII, and conclude this paper in Section VIII.

II. RELATED WORK

This section describes global control challenges and reviews highlights of HSI, block pushing, and compliant manipulation.

A. Global Control of Microrobots

This paper investigates global control of particles that have no onboard computation. This prevents us from applying controllers that require computation on the agents, as in [7]–[9]. Another control paradigm is to construct robots 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 in [10] and [11]; magnetic structures with different cross sections that can be independently steered in [12] and [13]; *MagMite* microrobots with different resonant frequencies and a global magnetic field by [14]; and magnetically controlled nanoscale helical screws constructed to stop movement at different cutoff frequencies of a global magnetic field by [1] and [15]. Similarly, our previous work focused on exploiting inhomogeneity between robots [16], [17]. These control algorithms theoretically apply to any number of robots, even robotic continuums. However, all these works never controlled more than twelve robots at a time because process noise cancels the differentiating effects of robot inhomogeneity. We desire control algorithms that extend to many thousands of robots. Limited position control was achieved in [18] and our previous work [19], but both used robots commanded in their local coordinate frame. Our new submission focuses on a more common paradigm: particles commanded in a global coordinate frame.

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

- 1) Control—global inputs: Many micro- and nanorobotic systems, see [1]–[3], [10]–[15], and [20], 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: n differential-drive robots in a two-dimensional (2-D) workspace require $3n$ state variables. Even holonomic robots require $2n$ state variables. Numerous methods exist for measuring this state in microrobotics [1], [3], [6]. 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 increase at small scales 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, see [17], and at worst a matrix inversion, see [16]. 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.

B. Human–Swarm Interaction

Most humans are able to, with practice, steer a swarm of robots controlled by a global input. Prior to this paper, no algorithm existed. Using human input to learn how to control a dynamic system is a line of research with a rich history [21], [22]. This paper exploits insights gained from Swarm-Control.net, particularly the fact that having a swarm's mean and variance is sufficient for object manipulation through an obstacle field.

A user interface enabling an operator to maneuver a swarm of robots through a cluttered workspace by specifying the bounding prism for the swarm and then translating or scaling this prism is designed in [23]. This paper shares the concept of a global control input, but our robots have no onboard computation and cannot track a virtual boundary.

Human *fanout*, the number of robots a single human user could directly control is studied in [24]. 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 multirobot search task, where users could assign goals to robots. Their user interaction studies with simulated planar robots indicated a *fanout plateau* of about eight robots, with diminishing returns for more robots. They hypothesized that the location of this plateau is highly dependent on the underlying task. Indeed, this paper indicates that there are tasks without plateaus. Their research investigated robots with three levels of autonomy. We use robots without autonomy, corresponding with their first-level robots.

Several user studies compare methods for controlling large swarms of simulated robots, for example [25]–[27]. 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, the studies [25], [26],

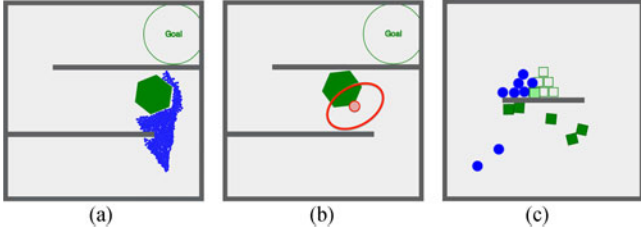


Fig. 2. Screenshots from our online experiments controlling multiparticle systems with limited, global control. (a) Varying the number of particles from 1 to 500. (b) Comparing four levels of visual feedback. (c) Varying noise from 0% to 200% of control authority.

and [27] were limited to a pool of 5, 18, and 32 participants. Using an online testing environment, we conduct similar studies but with sample sizes three orders of magnitude larger.

C. Block Pushing and Compliant Manipulation

Unlike *caging* manipulation, where robots form a rigid arrangement around an object, as in [28] and [29], our swarm of robots is unable to grasp the blocks they push, and so our manipulation strategies are similar to *nonprehensile manipulation* techniques, e.g., [30], where forces must be applied along the center of mass of the moveable object. A key difference is that our robots are compliant and tend to flow around the object, making this similar to fluidic trapping as in [31] and [32].

Our n -robot system with two control inputs and $4n$ states is inherently underactuated and superficially bears resemblance to compliant, underactuated manipulators. Our swarms conform to the object to be manipulated, but lack the restoring force provided by flexures in [33] or silicone in [34]. Our swarms tend to disperse and so to regroup them we require artificial forces like the variance control primitives in Section IV-C.

D. Relationship to Authors' Prior Work

This paper combines the content of two preliminary conference papers, extending their substance and providing full details in a single journal paper. One paper covered the first three months of SwarmControl.net experiments [35], and the second presented simulations of object manipulation [36]. This paper presents three years of results from SwarmControl.net. For object manipulation, this paper presents robust new algorithms for manipulation, path planning, and obstacle avoidance, and a rich set of parameter sweeps over key variables. All hardware validation experiments are new.

III. ONLINE EXPERIMENT

The goal of these online experiments is to test several scenarios involving large-scale HSI, and to do so with a statistically significant sample size. Towards this end, we have 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. 2. All code and experimental results are online in [37].

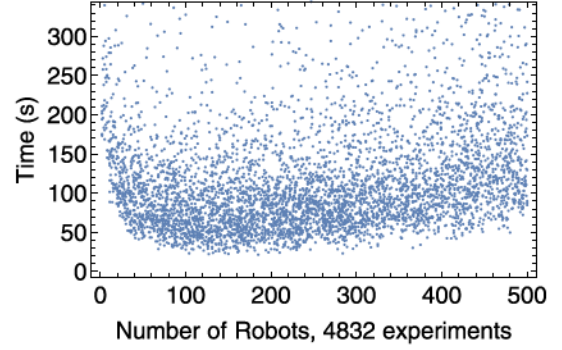


Fig. 3. Data from *Varying Number* using particles to push an object through a maze to a goal location.

We developed a flexible testing framework for online HSI studies. More than 5000 humans performed over 20 000 swarm-robotics experiments with this framework, logging almost 700 h of experiments. These experiments indicated three lessons used for designing automatic controllers for object manipulation with particle swarms.

- 1) When the number of particles is large (> 50), varying the number of particles does not significantly affect the performance.
- 2) Swarm control is robust to independent and identically distributed (IID) noise.
- 3) Controllers that only use the mean and variance of the swarm can perform better than controllers with full feedback.

A. Implementation

Our web server generates a unique identifier for each participant and sends it along with the landing page to the participant. A script on the participant's browser runs the experiment and posts the experiment data to the server. Anonymized human subject data were collected under IRB #14357-01.

We designed six experiments to investigate human control of large swarms for manipulation tasks. Screenshots of representative experiments are shown in Fig. 2. Each experiment examined the effects of varying a single parameter: population of particles for manipulation, four levels of visual feedback, different levels of Brownian noise. The users could choose which experiment to try, and our architecture randomly assigned a parameter value for each trial. We recorded the completion time and the participant ID for each successful trial.

B. Varying Number

This experiment varied from 1 to 500 the population of particles used to transport an object. The total area, maximum particle speed, and total net force the swarm could produce were constant. The particles pushed a large hexagonal object through an S-shaped maze. We hypothesized participants would complete the task faster with more particles. The results, shown in Fig. 3, do not support our hypothesis, indicating a minimum around 130 particles, but only a gradual increase in completion time from 50 to 500.

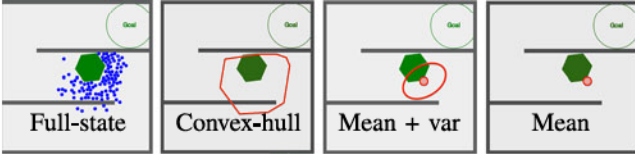


Fig. 4. Screenshots from task *Vary Visualization*. This experiment challenges players to quickly steer 100 particles (blue discs) to push an object (green hexagon) into a goal region. We record the completion time and other statistics.

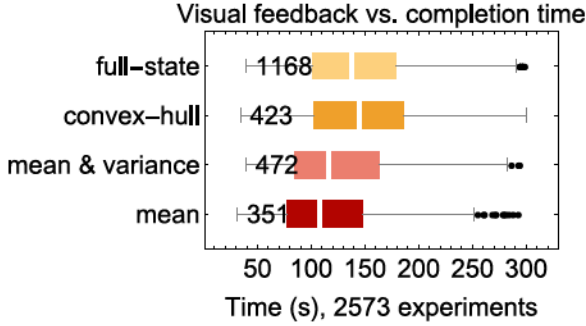


Fig. 5. Completion-time results for the four levels of visual feedback shown in Fig. 4. Players performed better with limited feedback.

C. Varying Visualization

This experiment explores manipulation with varying amounts of sensing information: *full-state* sensing provides the most information by showing the position of all particles; *convex hull* draws a convex hull around the outermost particles; *mean* provides the average position of the population; and *mean + variance* adds a confidence ellipse. Fig. 4 shows screenshots of the same particle swarm with each type of visual feedback. Full state requires $2n$ data points for n particles. Convex hull requires at worst $2n$, but according to Har-Peled [38], the expected number is $O(2n^{1/3})$. Mean requires two, and variance three, data points. Because they do not increase with population size, mean and mean + variance are convenient even with millions of particles.

Our hypothesis predicted a steady decrease in performance as the amount of visual feedback decreased. Our experiment indicated the opposite: players with just the mean completed the task faster than those with full-state feedback. As Fig. 5 shows, the levels of feedback arranged by increasing completion time are [mean, mean + variance, full state, convex hull]. All experiments lasting over 300 s were removed, under the assumption that the user stopped playing. Using ANOVA analysis, we rejected the null hypothesis that all visualization methods are equivalent, with p -value 2.69×10^{-19} . Anecdotal evidence from beta testers who played the game suggests that tracking 100 particles is overwhelming—similar to schooling phenomena that confuse predators—while working with just the mean + variance is like using a “spongy” manipulator. However, our beta testers described convex-hull feedback as confusing and irritating since it is not robust to outliers. A single particle left behind an obstacle will stretch the entire hull, obscuring the majority of the swarm.

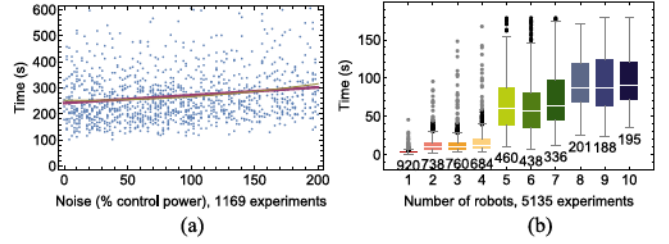


Fig. 6. (a) Varying the noise from 0% to 200% of the maximum control input resulted in only a small increase in completion time. (b) For position control, increasing the number of particles resulted in longer completion times. For more than four particles the goal pattern contained a void, which may have caused the jump in completion times.

D. Varying Noise

This experiment varied the strength of disturbances to study how noise (disturbance inputs) affects human control of large swarms. Noise was applied at every time step as follows:

$$\begin{aligned}\dot{x}_i &= u_x + m_i \cos(\psi_i) \\ \dot{y}_i &= u_y + m_i \sin(\psi_i).\end{aligned}$$

Here, m_i and ψ_i were uniformly IID, with $m_i \in [0, M]$ and $\psi_i \in [0, 2\pi]$. M was a constant for each trial ranging from 0% to 200% of the maximum control power ($u_{\max}$).

We hypothesized that 200% noise was the largest a human could be expected to control—at 200% noise, the particles move erratically. Disproving our hypothesis, the results in Fig. 6(a) show only a 40% increase in completion time for the maximum noise. This indicates that swarm control is robust to IID noise.

IV. GLOBAL CONTROL LAWS FOR A HOLONOMIC SWARM

Emboldened by the three lessons from our online experiments, this section presents automatic controllers for large numbers of particles that only rely on the first two moments of the swarm position distribution.

We represent particles as holonomic robots that move in the 2-D plane. We want to control position and velocity of the particles. First, assume a noiseless system containing one particle with mass m . Our inputs are global forces $[u_x, u_y]^T$. We define our state vector $\mathbf{x}(t)$ as the x position, x velocity, y position, and y velocity. The state-space representation in standard form is

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (1)$$

and our state-space representation is

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{1}{m} & 0 \\ 0 & 0 \\ 0 & \frac{1}{m} \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix}. \quad (2)$$

We want to find the number of states that we can control, which is given by the rank of the *controllability matrix*

$$\mathbf{C} = [\mathbf{B}, \mathbf{A}\mathbf{B}, \mathbf{A}^2\mathbf{B}, \mathbf{A}^3\mathbf{B}]. \quad (3)$$

$$\text{Here } C = \begin{bmatrix} 0 & 0 & \frac{1}{m} & 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{m} & 0 & 0 & 0 \\ 0 & \frac{1}{m} & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4)$$

$$\text{rank}(C) = 4 \quad (5) \quad \text{Thus}$$

and thus all four states are controllable. This section starts by proving independent position control of many particles is not possible, but the mean position can be controlled. We then provide conditions under which the variance of many particles is also controllable.

A. Independent Control of Many Particles is Impossible

In this model, a single particle is fully controllable. For holonomic particles, movement in the x and y coordinates are independent, so for notational convenience without loss of generality, we will focus only on movement in the x -axis. Given n particles to be controlled in the x -axis, there are $2n$ states: n positions and n velocities. Without loss of generality, assume $m = 1$. Our state-space representation is

$$\begin{bmatrix} \dot{x}_1 \\ \ddot{x}_1 \\ \vdots \\ \dot{x}_n \\ \ddot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & \dots & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ \dot{x}_1 \\ \vdots \\ x_n \\ \dot{x}_n \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u_x. \quad (6)$$

However, just as with one particle, we can only control two states because the controllability matrix C_n has rank two

$$C_n = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots \\ 1 & 0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \dots \\ 0 & 1 & 0 & 0 & \dots \\ 1 & 0 & 0 & 0 & \dots \end{bmatrix}, \quad \text{rank}(C_n) = 2. \quad (7)$$

B. Controlling the Mean Position

This means *any* number of particles controlled by a global command have just two controllable states in each axis. We cannot arbitrarily control the position and velocity of two or more particles, but have options on which states to control. We create the following reduced order system that represents the mean x position and velocity of the n particles

$$\begin{bmatrix} \dot{\bar{x}} \\ \ddot{\bar{x}} \end{bmatrix} = \frac{1}{n} \begin{bmatrix} 0 & 1 & \dots & 0 & 1 \\ 0 & 0 & \dots & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ \dot{x}_1 \\ \vdots \\ x_n \\ \dot{x}_n \end{bmatrix}$$

$$+ \frac{1}{n} \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u_x. \quad (8)$$

$$\begin{bmatrix} \dot{\bar{x}} \\ \ddot{\bar{x}} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{x} \\ \dot{\bar{x}} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_x. \quad (9)$$

We again analyze the controllability matrix C_μ

$$C_\mu = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \text{rank}(C_\mu) = 2. \quad (10)$$

Thus, the mean position and mean velocity are controllable.

There are several techniques for breaking the symmetry of the control input to allow controlling more states, for example by using obstacles as in [5], or by allowing independent noise sources as in [19].

We control mean position with a proportional derivative (PD) controller that uses the mean position and mean velocity. $[u_x, u_y]^T$ is the global force applied to each particle

$$\begin{aligned} u_x &= K_p(x_{\text{goal}} - \bar{x}) + K_d(0 - \dot{\bar{x}}) \\ u_y &= K_p(y_{\text{goal}} - \bar{y}) + K_d(0 - \dot{\bar{y}}). \end{aligned} \quad (11)$$

K_p is the proportional gain and K_d is the derivative gain.

C. Controlling the Variance

The variances σ_x^2 and σ_y^2 of n particles' position is

$$\begin{aligned} \bar{x}(\mathbf{x}) &= \frac{1}{n} \sum_{i=1}^n x_i, & \sigma_x^2(\mathbf{x}) &= \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 \\ \bar{y}(\mathbf{x}) &= \frac{1}{n} \sum_{i=1}^n y_i, & \sigma_y^2(\mathbf{x}) &= \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2. \end{aligned} \quad (12)$$

Controlling the variance requires being able to increase and decrease the variance. We will list a sufficient condition for each. Microscale systems are affected by unmodeled dynamics. These unmodeled dynamics are dominated by Brownian noise, as described in [39]. To model this, (1) must be modified as follows:

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) + W\boldsymbol{\varepsilon}(t) \quad (13)$$

where $W\boldsymbol{\varepsilon}(t)$ is a random perturbation produced by Brownian noise with magnitude W . Given a large obstacle-free workspace with $\mathbf{u}(t) = 0$, a *Brownian noise* process increases the variance linearly with time

$$\dot{\sigma}_x^2(\mathbf{x}(t), \mathbf{u}(t)) = W\boldsymbol{\varepsilon}, \quad \sigma_x^2(t) = \sigma_x^2(0) + W\boldsymbol{\varepsilon}t. \quad (14)$$

If faster dispersion is needed, the swarm can be pushed through obstacles such as a diffraction grating or Pachinko board as in [5].

If particles with radius r are in a bounded environment with sides of length $[\ell_x, \ell_y]$, the unforced variance asymptotically grows to the variance of a uniform distribution

$$[\sigma_x^2, \sigma_y^2] = \frac{1}{12}[(\ell_x - 2r)^2, (\ell_y - 2r)^2]. \quad (15)$$

A flat obstacle can be used to decrease variance. Pushing a group of dispersed particles against a flat obstacle will decrease their variance until the minimum-variance (maximum density) packing is reached. For large n , Graham and Sloane [40] showed that the minimum-variance packing for n circles with radius r is

$$\sigma_{\text{optimal}}^2(n, r) \approx \frac{\sqrt{3}}{\pi} nr^2 \approx 0.55 nr^2. \quad (16)$$

Thus, to control this variance, we choose

$$u(t) = \begin{cases} \text{move to wall} & \text{if } \sigma^2(\mathbf{x}) > \sigma_{\text{goal}}^2 \\ \text{move from wall} & \text{if } \sigma^2(\mathbf{x}) \leq \sigma_{\text{goal}}^2. \end{cases} \quad (17)$$

Similar to the PD controller in (11) that controls the mean particle position, a controller to regulate the variance to σ_{ref}^2 is

$$u_x = K_p(x_{\text{goal}}(\sigma_{\text{ref}}^2) - \bar{x}) - K_d \bar{v}_x + K_i(\sigma_{\text{ref}}^2 - \sigma_x^2) \quad (18)$$

$$u_y = K_p(y_{\text{goal}}(\sigma_{\text{ref}}^2) - \bar{y}) - K_d \bar{v}_y + K_i(\sigma_{\text{ref}}^2 - \sigma_y^2). \quad (19)$$

We call the gain scaling the variance error K_i because the variance, if unregulated, integrates over time. This controller requires a vertical and a horizontal wall. Equation (18) assumes the nearest wall is to the left of the particle at $x = 0$, and chooses a reference goal position such that the swarm, if uniformly distributed between 0 and ℓ , would have the correct variance according to (15)

$$x_{\text{goal}}(\sigma_{\text{ref}}^2) = \ell/2 = r + \sqrt{3\sigma_{\text{ref}}^2}. \quad (20)$$

If a wall to the right is closer, the signs of $[K_p, K_i]$ are inverted and the location x_{goal} is translated. A similar argument applies to (19).

D. Controlling Both Mean and Variance

The mean and variance of the swarm cannot be controlled simultaneously. However, if the variance gained while moving from a corner to the target position is less than some $\sigma_{\text{max}}^2 - \sigma_{\text{min}}^2$, we can adopt the hybrid, hysteresis-based controller shown in Algorithm 1 to regulate the mean and variance. Such a controller normally controls the mean position, but switches to minimizing variance if the variance exceeds σ_{max}^2 . Variance is reduced until less than σ_{min}^2 , then control again regulates the mean position. This technique satisfies control objectives that evolve at different rates as in [41], and the hysteresis avoids rapid switching between control modes. The process is illustrated in Fig. 7.

A key challenge is to select proper values for σ_{min}^2 and σ_{max}^2 . The optimal packing variance was given in (16). The random packings generated by pushing our particles into corners are

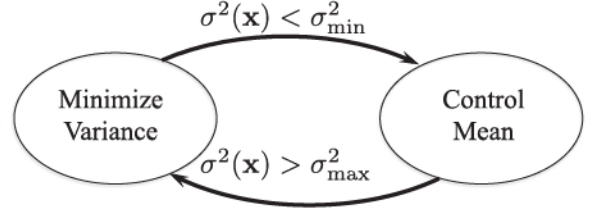


Fig. 7. Hysteresis to control swarm mean and variance.

Algorithm 1: Hybrid Mean and Variance Control..

Require: Knowledge of swarm mean $[\bar{x}, \bar{y}]$, variance $[\sigma_x^2, \sigma_y^2]$, the locations of the rectangular boundary $\{x_{\min}, x_{\max}, y_{\min}, y_{\max}\}$, and the target mean position $[x_{\text{target}}, y_{\text{target}}]$.

- 1: $x_{\text{goal}} \leftarrow x_{\text{target}}, y_{\text{goal}} \leftarrow y_{\text{target}}$
- 2: **loop**
- 3: **if** $\sigma_x^2 > \sigma_{\text{max}}^2$ **then**
- 4: $x_{\text{goal}} \leftarrow x_{\min}$
- 5: **else if** $\sigma_x^2 < \sigma_{\text{min}}^2$ **then**
- 6: $x_{\text{goal}} \leftarrow x_{\text{target}}$
- 7: **end if**
- 8: **if** $\sigma_y^2 > \sigma_{\text{max}}^2$ **then**
- 9: $y_{\text{goal}} \leftarrow y_{\min}$
- 10: **else if** $\sigma_y^2 < \sigma_{\text{min}}^2$ **then**
- 11: $y_{\text{goal}} \leftarrow y_{\text{target}}$
- 12: **end if**
- 13: Apply (11) to move toward $[x_{\text{goal}}, y_{\text{goal}}]$
- 14: **end loop**

suboptimal, so we choose the conservative values

$$\begin{aligned} \sigma_{\text{min}}^2 &= 2.5r + \sigma_{\text{optimal}}^2(n, r) \\ \sigma_{\text{max}}^2 &= 15r + \sigma_{\text{optimal}}^2(n, r). \end{aligned} \quad (21)$$

V. SIMULATION OF CONTROL LAWS

Our simulations use a Javascript port of Box2D, a popular 2-D physics engine with support for rigid-body dynamics, including collision, density, and friction, and fixed time step simulation [42]. All experiments in this section ran on a Chrome web browser on a 2.6-GHz Macbook. All code is available in [43].

1) Controlling the mean position:

We performed a parameter sweep using the PD controller (11) to identify the best control gains. Representative experiments are shown in Fig. 8. 100 particles were used and the maximum speed was 3 m/s. As shown in Fig. 8, we can achieve an overshoot of 1% and a rise time of 1.52 s with $K_p = 4$ and $K_d = 1$.

2) Controlling the variance:

Variance control uses the control law (18) with $K_{p,i,d} = [4, 1, 1]$. Results are shown in Fig. 9.

3) Hybrid control of mean and variance:

Fig. 10 shows a simulation run of the hybrid controller in Algorithm 1 with 100 particles in a square workspace containing no internal obstacles.

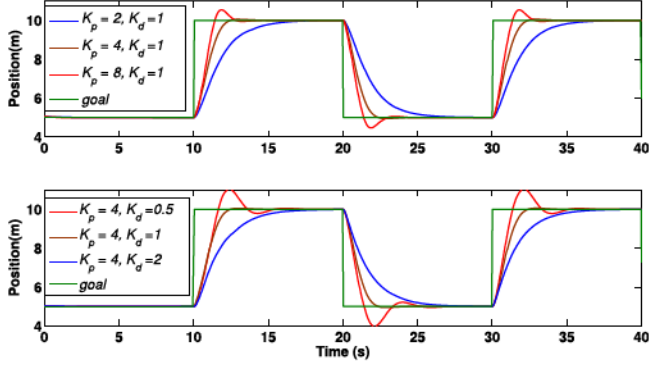


Fig. 8. In simulation, tuning proportional (K_p , top) and derivative (K_d , bottom) gain values in (11) improves performance with $n = 100$ particles.

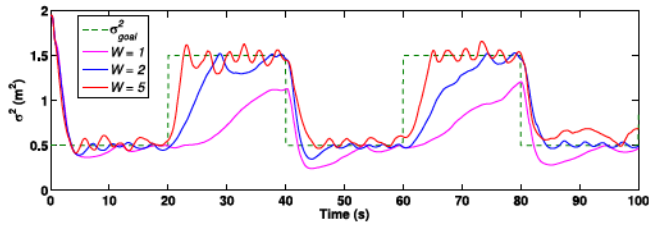


Fig. 9. In simulation, increased noise results in more responsive variance control because stronger Brownian noise causes a faster increase of variance.

TABLE I
SUMMARY, OBJECT MANIPULATION RESULTS

Method & Results (10 trials)	mean $\pm$ std (s)
Value iteration (VI)	367 $\pm$ 253
VI + potential fields (PF)	271 $\pm$ 267
VI + outlier rejection (OR)	245 $\pm$ 135
BFS + PF + OR	183 $\pm$ 179
VI + PF + OR	90 $\pm$ 35

VI. PARTICLE SWARM OBJECT MANIPULATION

This section analyzes an *object manipulation* task attempted by our hybrid, hysteresis-based controllers. The swarm must deliver the object to the goal region. We assume Coulomb and viscous friction parameters such that the object can be moved by particle motion. Increasing the number of pushing particles increases the object speed. To solve this object manipulation task, we divide the task into three components: 1) designing a policy for the object; 2) pushing the object with a compliant swarm; and 3) managing outliers.

Table I summarizes results for successful simulation trials.

A. Learning a Policy for the Object

To design the policy, we first discretize the environment. In [36], we used breadth-first search (BFS) on this discretized grid, but using workspace BFS fails to account for the hull of the object and will suggest moves that can cause collisions with the workspace. A configuration-space BFS approach avoids that problem but still fails to model uncertain actuation of the object by the swarm.

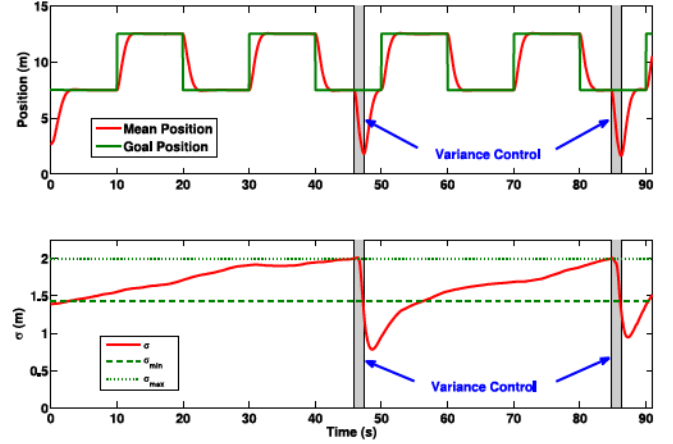


Fig. 10. Simulation result with 100 particles under hybrid control Algorithm 1, which controls both the mean position (top) and variance (bottom). For ease of analysis, only x position and variance are shown.

To solve both these problems, this paper models object movement as a Markov decision process (MDP) with nondeterministic movement. Value iteration is used to learn an *optimal policy* [44]. At each state, the object can be commanded to move in one of eight directions with a small probability of moving in a wrong direction.

The reward function $r(x, u)$ is defined as

$$r(x, u) = \begin{cases} +100, & \text{if } u \text{ leads to goal state} \\ -100, & \text{if } u \text{ leads to an obstacle state} \\ -1, & \text{otherwise} \end{cases} \quad (22)$$

where x is the current state and u is the action. Value iteration computes $\hat{V}(x)$, the expected discounted sum reward if the optimal policy is implemented, for the object starting in each state x . The optimal policy is

$$D(x) = \arg \max_u \left[r(x, u) + \sum_{j=1}^N \hat{V}(x_j) p(x_j | x, u) \right]. \quad (23)$$

The value function $\hat{V}(x_i)$ is calculated by computing the value $\hat{V}$ for all N states and iterating until convergence

for $i = 1$ to N do

$$\hat{V}(x_i) = \gamma \max_u \left[r(x_i, u) + \sum_{j=1}^N \hat{V}(x_j) p(x_j | x_i, u) \right]$$

end. (24)

Our probabilistic motion model $p(x_j | x, u)$ assumed that the object moved in the commanded direction u half of the time but $+45^\circ$ with probability 0.25 and -45° with probability 0.25. In our experiments, $\gamma = 0.97$ and (24) was iterated 200 times.

M_{BFS} and the value function are shown in Fig. 11. In 10 simulations with 100 particles, pushing the object to goal using BFS required 183 ± 179 s, whereas value iteration required 90 ± 35 s (mean $\pm$ std).

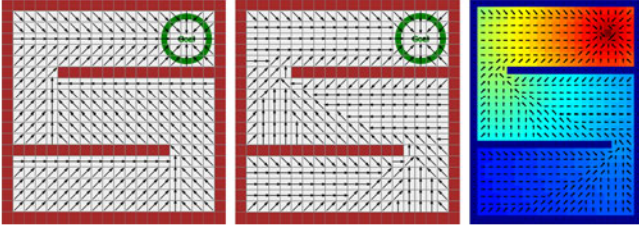


Fig. 11. BFS finds the shortest path for the moveable object to compute gradient vectors (left). Modeling as an MDP enables encoding penalties for being near obstacles. (Middle) Control policy from value iteration. (Right) Vision algorithm detects obstacles in the hardware setup. This map is used to produce the value function and control policy shown.

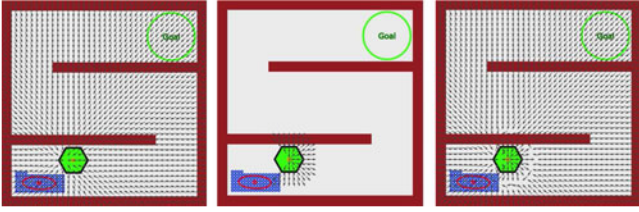


Fig. 12. (Left) Attractive field is centered behind the object's COM. (Middle) Repulsive field is centered at the object's COM. (Right) Combining these forces prevents the swarm from pushing the object backwards.

B. Potential Fields for Swarm Management With a Compliant Manipulator

When the swarm is in front of the object, control law (11) pushes the object backwards. To fix this, we implement a potential field approach that attracts the swarm to the intermediate goal, but repulses the swarm from in front of the object, as shown in Fig. 12. The approach is similar to [46, Ch. 5]. The repulsive potential field is centered at b , the object's COM, and is active in a circular sector of angular width 2θ and radius d_0 aligned with $D(b)$. $D(b)$ is the desired direction of motion from (23)

$$\begin{aligned} d &= [\bar{x}, \bar{y}] - b, & \phi &= \cos^{-1} \left(\frac{D(b) \cdot d}{\|D(b)\| \cdot \|d\|} \right) \\ F_{\text{att}} &= -\zeta \frac{d}{\|d\|} \\ F_{\text{rep}} &= \begin{cases} \eta \left(\frac{1}{\|d\|} - \frac{1}{d_0} \right) \frac{1}{\|d\|^2} d, & \|d\| \leq d_0 \wedge \phi < \theta \\ 0, & \text{otherwise} \end{cases} \\ F_{\text{pot}} &= F_{\text{att}} + F_{\text{rep}} \\ [x_{\text{goal}}, y_{\text{goal}}] &= [\bar{x}, \bar{y}] + \frac{F_{\text{pot}}}{\|F_{\text{pot}}\|}. \end{aligned} \quad (25)$$

Here, η and ζ are positive parameters that scale the forces and $\|d\|$ is the distance from the swarm mean $[\bar{x}, \bar{y}]$ to the object COM. In simulations, $\theta = \pi/2$, $\eta = 75$, $\zeta = 2$, and $d_0 = 3$. Because the kilobots have a slower time constant, they use $\theta = \pi/2$, $\eta = 50$, $\zeta = 1$, and $d_0 = 7.5$.

In 10 simulations with 100 particles, pushing the object to goal without a repulsive potential field failed in 2 of 12 runs. No failures occurred with the repulsive potential field. Of successful trials, completion time without repulsive potential fields required 245 ± 135 s, whereas using repulsive potential fields required 90 ± 35 s (mean $\pm$ std).

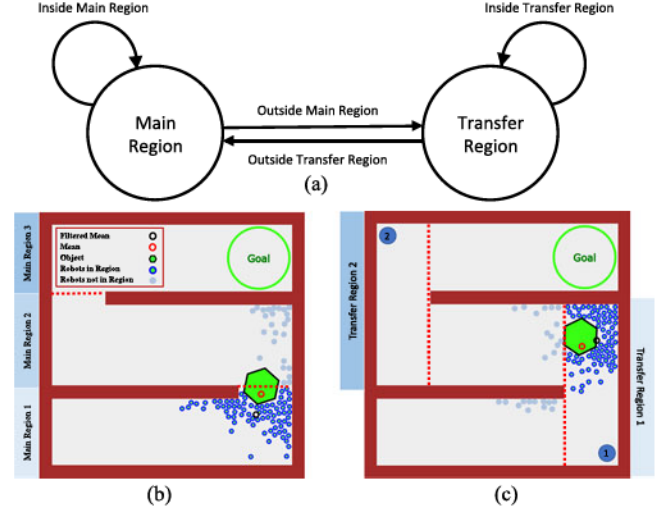


Fig. 13. Outlier rejection state machine and regions.

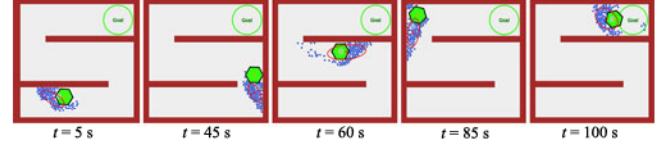


Fig. 14. Snapshots showing an object manipulation simulation with 100 particles under automatic control (see also Extension 1).

C. Outlier Rejection

The variance controller in Algorithm 1 is a greedy algorithm that is susceptible to outliers. Our controller in [36] failed in 14% trials, in each failure some particles were unable to reach the object because workspace obstacles were blocking them. This failure rate increases if object weight increases or ground-particle friction increases. The mean and covariance calculations (12) included all particles in the workspace. Particles that cannot reach the object due to obstacles skew these calculations. The state machine in Fig. 13(a) solves this problem by creating two states for the maze: either main or transfer. Each state has a set of regions representing a discretized visibility polygon. Whenever the object crosses a region boundary, the state toggles. The *main* regions are generated by extending obstacles until they meet another obstacle. The *transfer* regions are perpendicular to obstacle boundaries and act as a buffer between two main regions.

Fig. 13(b) shows the regions for the main state. The object is in region 1. An indicator function is applied to (12) so only particles inside region 1 are counted. This filtering increases experimental success because the mean calculation only includes nearby particles that can directly interact with the object. When the object leaves main region 1, the state switches to transfer. The transfer regions are shown in Fig. 13(c). The object is in transfer region 1, so only particles in transfer region 1 are included in the mean and covariance calculations. The particles should push the object to the left. Without filtering using regions, the red circle is the mean and the algorithm would instruct the particles to push the object up. The black circle shows the filtered mean and the algorithm instructs the particles to push the object directly left.

Algorithm 2: Object-manipulation Controller for a Particle Swarm.

require: Knowledge of moveable object's center of mass b ;
 swarm mean $[\bar{x}, \bar{y}]$ and variance $[\sigma_x^2, \sigma_y^2]$, each
 calculated using the regions function from §VI-C;
 map of the environment

```

1: Compute optimal policy for object, according to §VI-A
2: while  $b$  is not in goal region do
3:    $\sigma^2 \leftarrow \max(\sigma_x^2, \sigma_y^2)$ 
4:   if  $\sigma^2 > \sigma_{\max}^2$  then
5:     while  $\sigma^2 > \sigma_{\min}^2$  do
6:        $[x_{\text{goal}}, y_{\text{goal}}] \leftarrow$  nearest corner in region
7:       Apply (11) to move toward  $[x_{\text{goal}}, y_{\text{goal}}]$ 
8:     end while
9:   else
10:    Calculate  $D(b)$   $\triangleright$  direction for object at  $b$ 
11:    Apply (25)  $\triangleright$  potential field for swarm
12:   end if
13: end while

```

In 10 simulations with 100 particles, completion time without outlier rejection required 271 ± 267 s, whereas using outlier rejection required 90 ± 35 s (mean $\pm$ std).

D. Simulation Results

We use the hybrid hysteresis-based controller in Algorithm 1 to track the desired position while maintaining sufficient particle density to move the object by switching to minimize variance whenever variance exceeds a set limit: $0.003W$ and $0.006W$ were added to the min and max variance limits from (21), where W is the magnitude of the Brownian noise. The minimize variance control law (18) is slightly modified to choose the nearest corner further from the goal than the object with an obstacle-free straight-line path to the object. The control algorithm for object manipulation is listed in Algorithm 2.

In rare cases, during simulations, the swarm may become trapped in a local minimum of (25). If the swarm mean position does not change for 5 s, the swarm is assumed to be in a local minimum and is commanded to move toward the previous corner. As soon as the mean position changes, normal control resumes.

Fig. 14 shows snapshots during an execution of this algorithm in simulation. Fig. 16(a) shows the six equal-area object shapes tested in simulations. To illustrate the flexibility of the algorithm, we tested two additional workspaces, *E-shaped* and *Spiral*, without changing the algorithm. These are shown in Fig. 15. More complicated workspaces could be generated by composing these workspaces. Fig. 16(b) shows the results of all three mazes. The E-shaped maze required the least average time because the path to the goal is shorter. Experimental results of parameters sweeps are summarized in Fig. 17. Each trial measured the time to deliver the object to the goal location. The default parameter settings used 100 particles, a normalized weight of 1, a hexagon shape, and Brownian noise (applied once each simulation step) with $W = 5$.

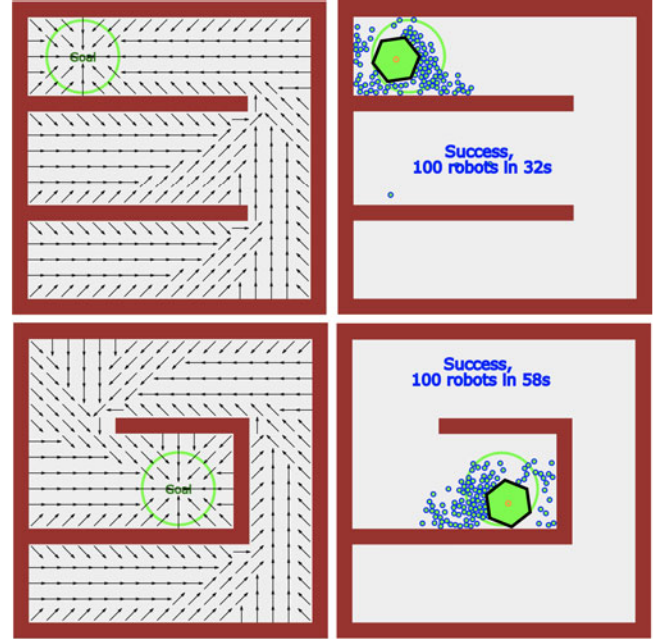


Fig. 15. We tested three workspaces. The control policies from value iteration for an E-shaped and a spiral workspace are shown in the left column.

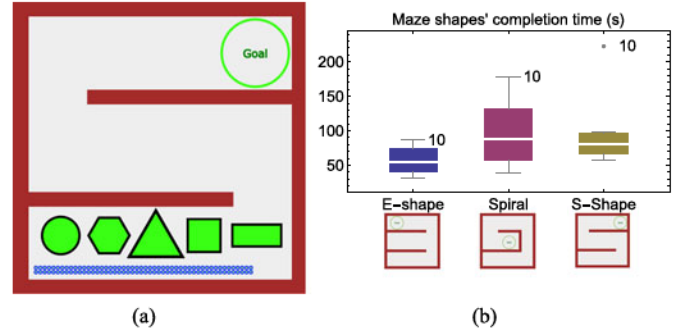


Fig. 16. (a) The six equal-area objects tested in simulation. (b) Completion times for three workspaces.

The interaction between the particles and object is impulsive so, like the study of impulsive pulling in [47], adding additional particles decreases completion time, but with diminishing returns. The effect of adding particles diminishes asymptotically because additional particles have difficulty interacting with the object. Brownian noise adds stochasticity. This randomness can break the object free if it is stuck, but diminishes the effect of the control input. Increasing noise increases completion time. The particles have limited force, so increasing the object weight increases completion time. Each shape was designed to have the same mass and area. Rectangles and squares tend to get stuck in the 90° workspace corners, and cause longer completion times than circles, triangles, and hexagons.

VII. OBJECT MANIPULATION WITH HARDWARE ROBOTS

Our experiments use centimeter-scale hardware systems called *kilobots*. While those are far larger than the microscale devices we model, using kilobots allows us to emulate a variety of dynamics, while enabling a high degree of control over robot function, the environment, and data collection. The kilobot is

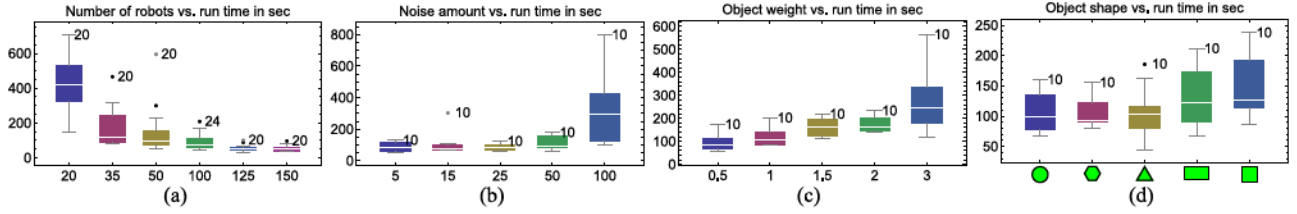


Fig. 17. Parameter sweep simulation studies for (a) number of particles, (b) different noise values, (c) object weight, and (d) object shape. Each bar is labeled with the number of trials. Completion time is in seconds.



Fig. 18. Hardware platform. At right are the shapes used for hardware experiments and a visualization of the potential field.

a nonholonomic, low-cost robot designed for testing collective algorithms with large numbers of robot [48], [49]. It is available as an open-source platform or commercially [50]. Each robot is approximately 3 cm in diameter, 3-cm tall, and uses two vibration motors to move on a flat surface at speeds up to 1 cm/s. Each robot has one ambient light sensor that is used to implement *phototaxis*, moving towards a light source.

A. Environmental Setup

In these experiments, as shown in Fig. 18, we used $n = 100$ kilobots and a $1.5 \text{ m} \times 1.2 \text{ m}$ whiteboard as the workspace. LED floodlights were placed 1.5 m above the table and corners of a square with 6-m sides. An Arduino Uno connected to an eight-relay shield controlled the lights.

Above the table, an overhead machine vision system tracks the swarm. The vision system identifies obstacles and the object by color segmentation, determines the corners of the maze, and identifies robots using a circular Hough transform.

The objects were 3-D printed from acrylonitrile butadiene styrene (ABS) plastic with a paper overlay. Shapes included a 325-cm^2 equilateral triangle, 324-cm^2 square, 281-cm^2 hexagon, 254-cm^2 circle, and a 486-cm^2 rectangle, all shown in Fig. 18. The laser-cut patterns for the neon green fiducial markers on the robots and 3-D files for objects are available at our github repository [43].

- 1) *Swarm mean control (hardware experiment)*: Unlike the PD controller (11), we cannot command a force input to the kilobots. Instead, control is given by turning on one of eight lights. The kilobots run a phototaxis routine where they search for an orientation that aligns them with the light source, and then move with an approximately

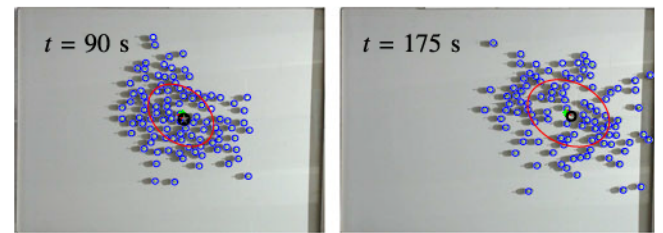
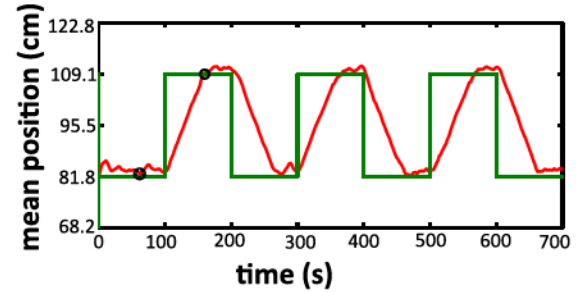


Fig. 19. Regulating mean x position of 100 kilobots using control law (11).

constant velocity toward this light. The kilobots oscillate along this orientation because they only have one light detector.

We use the sign of (11), and choose the closest orientation to D(b) among the eight light sources. Fig. 19 shows that this limited, discretized control still enables regulating the mean position of a swarm of 100 robots.

B. Automated Object Manipulation (Hardware Experiment)

Even though kilobots are nonholonomic, they performed five successful runs manipulating a hexagonal object through an

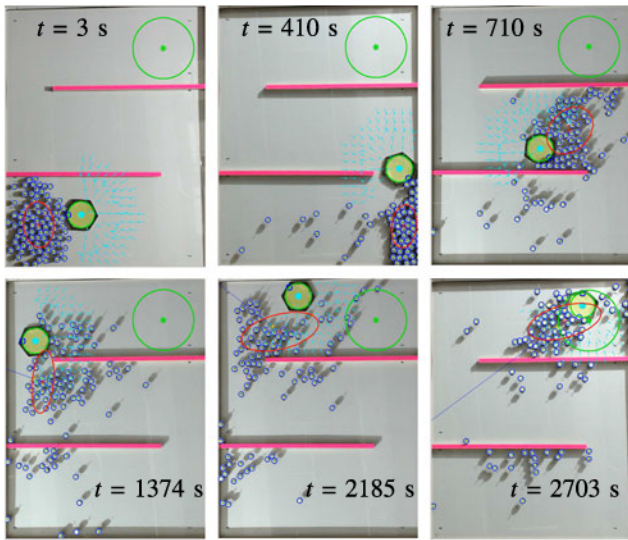


Fig. 20. Snapshots showing object manipulation experiment with 100 kilobots under automatic control. The automatic controller generates a policy to the goal (see Extension 2).

obstacle maze. Videos of these runs are in Extension 2. These hardware experiments represent the results of more than 100 h of trials. Each trial used 100 kilobots. Trials two through five were performed in a row with no failures in between. For each trial, fully charged kilobots were placed in the lower left hand of the workspace, as shown in Fig. 20. The moveable object was placed in the lower center of the workspace. MATLAB code for vision processing, the value iteration of Section VI-A, and the algorithm of Section VI-D is available on MATLAB Central in [51]. Trials were run until the object COM entered the goal region. The trials ran for {1465, 3457, 3000, 2162, 2707} s. This is 2558 ± 771 s (mean $\pm$ std).

We also tested other object shapes. A circular object completed in 3155 s. A square object completed in 6871 s. A rectangle and three equilateral triangle objects of varying sizes failed in a total of nine runs. Manipulation failures occurred when the object was pushed into a corner, requiring torque to be unstuck. Swarm torque control is the subject of our ongoing research being in [52].

VIII. CONCLUSION

The small size of micro- and nanoparticles makes individual control and autonomy challenging, so currently these particles are steered by global control inputs such as magnetic fields or chemical gradients. To investigate this control challenge, this paper introduced SwarmControl.net, an open-source tool for large-scale user experiments where human users steer swarms of robots to accomplish tasks. Analysis of the gameplay results revealed benefits of measuring and controlling statistics of the swarm rather than full-state feedback, robustness to IID noise, and small effects of varying population size of large swarms.

Inspired by the three lessons from SwarmControl.net, this paper designed controllers and controllability results using only the mean and variance of a particle swarm. We developed a

hysteresis-based controller to regulate the position and variance of a swarm. We designed a controller for object manipulation using value iteration for path planning, regions for outlier rejection, and potential fields for minimizing moving the object backwards. All automatic controllers were implemented using 100 kilobots steered by the direction of a global light source. These experiments culminated in an object manipulation task in a workspace with obstacles.

Our future goal is to perform assembly using particle swarms to manipulate and attach components. This task requires applying force and torque to components and manipulating them through obstacles and each other. This paper provides foundational algorithms and techniques for steering swarms, object manipulation, and addressing obstacle fields, but there are many opportunities to extend the work.

Topics of interest include control with nonuniform flow such as fluid in an artery, gradient control fields like that of an MRI, competitive playing, multimodal control, flexible workspaces, optimal control, and targeted drug delivery in a vascular network.

ACKNOWLEDGMENT

The authors would like thank the anonymous reviewers and D. Shell for their substantive comments that improved the presentation and content of this paper.

REFERENCES

- [1] K. E. Peyer, L. Zhang, and B. J. Nelson, "Bio-inspired magnetic swimming microrobots for biomedical applications," *Nanoscale*, vol. 5, no. 4, pp. 1259–1272, 2013.
- [2] Y. Shirai, A. J. Osgood, Y. Zhao, K. F. Kelly, and J. M. Tour, "Directional control in thermally driven single-molecule nanocars," *Nano Lett.*, vol. 5, no. 11, pp. 2330–2334, Feb. 2005.
- [3] P.-T. Chiang *et al.*, "Toward a light-driven motorized nanocar: Synthesis and initial imaging of single molecules," *ACS Nano*, vol. 6, no. 1, pp. 592–597, Feb. 2011.
- [4] K. Sugawara, N. Correll, and D. Reishus, "Object transportation by granular convection using swarm robots," in *Distributed Autonomous Robotic Systems*. New York, NY, USA: Springer, 2014, pp. 135–147.
- [5] A. T. Becker, G. Habibi, J. Werfel, M. Rubenstein, and J. McLurkin, "Massive uniform manipulation: Controlling large populations of simple robots with a common input signal," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 520–527.
- [6] S. Martel, S. Taherkhani, M. Tabrizian, M. Mohammadi, D. de Lanauze, and O. Felfoul, "Computer 3d controlled bacterial transports and aggregations of microbial adhered nano-components," *J. Micro-Bio Robot.*, vol. 9, no. 1–2, pp. 23–28, 2014.
- [7] D. Milutinovic and P. Lima, "Modeling and optimal centralized control of a large-size robotic population," *IEEE Trans. Robot.*, vol. 22, no. 6, pp. 1280–1285, Dec. 2006.
- [8] A. Prorok, N. Correll, and A. Martinoli, "Multi-level spatial modeling for stochastic distributed robotic systems," *Int. J. Robot. Res.*, vol. 30, no. 5, pp. 574–589, 2011.
- [9] N. Demir and B. Açıkmeşe, "Probabilistic density control for swarm of decentralized on-off agents with safety constraints," in *Proc. 2015 IEEE Amer. Control Conf.*, 2015, pp. 5238–5244.
- [10] B. R. Donald, C. G. Levey, C. D. McGray, I. Paprotny, and D. Rus, "An untethered, electrostatic, globally controllable MEMS micro-robot," *J. Microelectromech. Syst.*, vol. 15, no. 1, pp. 1–15, Feb. 2006.
- [11] B. R. Donald, C. G. Levey, and I. Paprotny, "Planar microassembly by parallel actuation of MEMS microrobots," *J. Microelectromech. Syst.*, vol. 17, no. 4, pp. 789–808, Aug. 2008.
- [12] S. Floyd, E. Diller, C. Pawashe, and M. Sitti, "Control methodologies for a heterogeneous group of untethered magnetic micro-robots," *Int. J. Robot. Res.*, vol. 30, no. 13, pp. 1553–1565, Nov. 2011.

- [13] E. Diller, J. Giltinan, and M. Sitti, "Independent control of multiple magnetic microrobots in three dimensions," *Int. J. Robot. Res.*, vol. 32, no. 5, pp. 614–631, 2013. [Online]. Available: <http://ijr.sagepub.com/content/32/5/614.abstract>
- [14] D. Frutiger, B. Kratochvil, K. Vollmers, and B. J. Nelson, "Magmites—Wireless resonant magnetic microrobots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Pasadena, CA, USA, May 2008, pp. 1770–1771.
- [15] S. Tottori, L. Zhang, F. Qiu, K. K. Krawczyk, A. Franco-Obregón, and B. J. Nelson, "Magnetic helical micromachines: Fabrication, controlled swimming, and cargo transport," *Adv. Mater.*, vol. 24, no. 811, pp. 811–816, 2012.
- [16] A. T. Becker and T. Bretl, "Approximate steering of a unicycle under bounded model perturbation using ensemble control," *IEEE Trans. Robot.*, vol. 28, no. 3, pp. 580–591, Jun. 2012.
- [17] A. T. Becker, C. Onyuksel, and T. Bretl, "Feedback control of many differential-drive robots with uniform control inputs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 2256–2262.
- [18] T. Bretl, "Control of many agents using few instructions," *Robot. Sci. Syst.*, vol. 3, pp. 209–216, 2007.
- [19] A. T. Becker, C. Onyuksel, T. Bretl, and J. McLurkin, "Controlling many differential-drive robots with uniform control inputs," *Int. J. Robot. Res.*, vol. 33, no. 13, pp. 1626–1644, 2014.
- [20] K. Takahashi, N. Ogawa, H. Oku, and K. Hashimoto, "Organized motion control of a lot of microorganisms using visual feedback," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2006, pp. 1408–1413.
- [21] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robot. Auton. Syst.*, vol. 57, no. 5, pp. 469–483, 2009.
- [22] P. Abbeel, A. Coates, and A. Y. Ng, "Autonomous helicopter aerobatics through apprenticeship learning," *Int. J. Robot. Res.*, vol. 29, no. 13, pp. 1608–1639, 2010.
- [23] N. Ayanian, A. Spielberg, M. Arbesfeld, J. Strauss, and D. Rus, "Controlling a team of robots with a single input," in *Proc. 2014 IEEE Int. Conf. Robot. Autom.*, 2014, pp. 1755–1762.
- [24] D. R. Olsen, Jr., and S. B. Wood, "Fan-out: Measuring human control of multiple robots," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, Vienna, Austria, Apr. 2004, pp. 231–238.
- [25] S. Bashyal and G. K. Venayagamoorthy, "Human swarm interaction for radiation source search and localization," in *Proc. IEEE Swarm Intell. Symp.*, 2008, pp. 1–8.
- [26] J.-P. de la Croix and M. Egerstedt, "Controllability characterizations of leader-based swarm interactions," in *Proc. AAAI Fall Symp. Series*, 2012, pp. 1–6.
- [27] A. Kolling, S. Nunnally, and M. Lewis, "Towards human control of robot swarms," in *Proc. 7th Annu. ACM/IEEE Int. Conf. Hum.-Robot Interact.*, 2012, pp. 89–96.
- [28] A. Sudsang, F. Rothganger, and J. Ponce, "Motion planning for disc-shaped robots pushing a polygonal object in the plane," *IEEE Trans. Robot. Autom.*, vol. 18, no. 4, pp. 550–562, Aug. 2002.
- [29] J. Fink, M. A. Hsieh, and V. Kumar, "Multi-robot manipulation via caging in environments with obstacles," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 1471–1476.
- [30] K. M. Lynch, "Locally controllable manipulation by stable pushing," *IEEE Trans. Robot. Autom.*, vol. 15, no. 2, pp. 318–327, Apr. 1999.
- [31] M. D. Armani, S. V. Chaudhary, R. Probst, and B. Shapiro, "Using feedback control of microflows to independently steer multiple particles," *J. Microelectromech. Syst.*, vol. 15, no. 4, pp. 945–956, Aug. 2006.
- [32] A. T. Becker, R. Sandheinrich, and T. Bretl, "Automated manipulation of spherical objects in three dimensions using a gimbaled air jet," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 781–786.
- [33] L. U. Odhner *et al.*, "A compliant, underactuated hand for robust manipulation," *Int. J. Robot. Res.*, vol. 33, no. 5, pp. 736–752, 2014.
- [34] R. Deimel and O. Brock, "A novel type of compliant, underactuated robotic hand for dexterous grasping," in *Proc. Robot. Sci. Syst.*, Berkeley, CA, USA, 2014, pp. 1687–1692.
- [35] A. T. Becker, C. Ertel, and J. McLurkin, "Crowdsourcing swarm manipulation experiments: A massive online user study with large swarms of simple robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 2825–2830.
- [36] S. Shahrokhi and A. T. Becker, "Stochastic swarm control with global inputs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2015, pp. 421–427.
- [37] C. Ertel, S. Shahrokhi, and A. T. Becker, "SwarmControl.net git repository," Aug. 2016. [Online]. Available: <https://github.com/RoboticSwarmControl/SwarmControlRedux.git>
- [38] S. Har-Peled, "On the expected complexity of random convex hulls," arXiv:1111.5340, 2011.
- [39] A. Einstein, *Investigations on the Theory of the Brownian Movement*. North Chelmsford, MA, USA: Courier Corporation, 1956.
- [40] R. L. Graham and N. J. Sloane, "Penny-packing and two-dimensional codes," *Discr. Comput. Geom.*, vol. 5, no. 1, pp. 1–11, 1990.
- [41] M. Kloetzer and C. Belta, "Temporal logic planning and control of robotic swarms by hierarchical abstractions," *IEEE Trans. Robot.*, vol. 23, no. 2, pp. 320–330, Apr. 2007.
- [42] E. Catto, "User manual, Box2D: A 2D physics engine for games," 2010. [Online]. Available: <http://www.box2d.org>
- [43] S. Shahrokhi, M. Wan, L. Lin, and A. T. Becker, "Steering swarm simulation," Aug. 2016. [Online]. Available: <http://goo.gl/qsVqTU>
- [44] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. Cambridge, MA, USA: MIT Press, Sep. 2005.
- [45] A. T. Becker, "MDP robot grid-world example," [Online]. Available: <https://www.mathworks.com/matlabcentral/fileexchange/49992-mdp-robot-grid-w>
- [46] M. W. Spong and M. Vidyasagar, *Robot Dynamics and Control*. Hoboken, NJ, USA: Wiley, 2008.
- [47] D. L. Christensen, S. A. Suresh, K. Hahm, and M. R. Cutkosky, "Let's all pull together: Principles for sharing large loads in microrobot teams," *IEEE Robot. Autom. Lett.*, vol. 1, no. 2, pp. 1089–1096, Jul. 2016.
- [48] M. Rubenstein, C. Ahler, and R. Nagpal, "Kilobot: A low cost scalable robot system for collective behaviors," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 3293–3298.
- [49] M. Rubenstein, A. Cornejo, and R. Nagpal, "Programmable self-assembly in a thousand-robot swarm," *Science*, vol. 345, no. 6198, pp. 795–799, 2014.
- [50] K-Team, "Kilobot," 2015. [Online]. Available: www.k-team.com/mobile-robotics-products/kilobot
- [51] S. Shahrokhi, L. Lin, M. Wan, and A. T. Becker, "Kilobot swarm control using Matlab + Arduino," Aug. 2016. [Online]. Available: <http://mathworks.com/matlabcentral/fileexchange/58690>
- [52] S. Shahrokhi and A. T. Becker, "Object manipulation and position control using a swarm with global inputs," in *Proc. IEEE Conf. Autom. Sci. Eng.*, Aug. 2016, pp. 1–6.



Shiva Shahrokhi (S'14) received the B.S. degree in computer engineering from Iran University of Science and Technology, Tehran, Iran. She is currently working toward the Ph.D. degree in the Electrical and Computer Engineering Department, University of Houston, Houston, TX, USA.

She is involved in studies of swarm manipulation and control for robotics applications. These include steering massively underactuated mobile robotic ensembles, swarm shape control, and machine learning.



Lillian Lin (S'14) is currently working toward the B.S. degree in electrical engineering from University of Houston, Houston, TX, USA.

She is the President of the student IEEE branch on campus. She is an Undergraduate Researcher with the Robotics Swarm Control Laboratory, University of Houston. Her main research interests include robotic control systems.



Chris Ertel received the B.S. degree in mechanical engineering from Rice University, Houston, TX, USA, in 2010.

He has worked in industry and startups developing games, computer-aided design applications, medical telemetry systems, and other software. Since 2013 he has collaborated with Dr. Becker on SwarmControl.net and related projects at University of Houston Robotic Swarm Control Laboratory and the Rice Multi-Robot Systems Laboratory.



Mable Wan received the B.S. degree in electrical and computer engineering and the B.B.A. degree in entrepreneurship from University of Houston, Houston, TX, USA, in 2017.

Since May 2017, she has been a software engineer with Northrop Grumman Corporation, Baltimore, MD, USA.



Aaron T. Becker (S'06–M'12–SM'17) received the M.S. and Ph.D. degrees in electrical and computer engineering from University of Illinois at Urbana-Champaign, Champaign, IL, USA, in 2008 and 2012, respectively.

He was a Postdoctoral Research Scholar with the Multi-Robot Systems Laboratory, Rice University, and a Postdoctoral Research Fellow with Boston Children's Hospital, Harvard Medical School, Boston, MA, USA, before joining the Electrical and Computer Engineering Department, University of

Houston, Houston, TX, USA, as an Assistant Professor in 2014. He is active in developing control laws and algorithms for robot swarms.

He received the Best Paper Award at IROS 2013 and the NSF CAREER Award in 2016.