Stochastic Swarm Control with Global Inputs

Shiva Shahrokhi and Aaron T. Becker

***Abstract*— Micro- and nanorobots can be built in large numbers, but generating independent control inputs for each robot is prohibitively difficult. Instead, micro- and nanorobots are often controlled by a global field. In previous work we conducted large-scale human-user experiments where humans played games that steered large swarms of simple robots to complete tasks such as manipulating blocks. One surprising result was that humans completed a block-pushing task *faster* when provided with only the mean and variance of the robot swarm than with full-state feedback. Inspired by human operators, this paper investigates controllers that use only the mean and variance of a robot swarm. We prove that the mean position is controllable, and show how an obstacle can make the variance controllable. We next derive automatic controllers for these and a hybrid, hysteresis-based switching control to regulate the first two moments of the robot distribution. Finally, we employ these controllers as primitives for a block-pushing task.**

I. Introduction

Micro- and nanorobotics have diverse potential applications in targeted material delivery, construction, assembly, and surgery. Constraints on computation prevent autonomous operation, and direct control over individual units scales poorly with population size. Instead these systems often use global control signals broadcast to the entire robot population. Additionally, it is not always possible to gather pose information on each robot for feedback control. Robots might be difficult or impossible to sense individually due to their size and location. For example, micro-robots are smaller than the minimum resolution of a clinical MRI-scanner [1]. However, it is often possible to sense global 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 requiring only mean and variance measurements of the robot's positions. These controllers are used as primitives to perform a block-pushing task illustrated in Fig. 1.

II. Related Work

A. Global-control of micro- and nanorobots

We are particularly motivated by harsh constraints in micro- and nanorobotic systems. Small robots are often powered and steered by a global, broadcast control signal. Examples include *scratch-drive microrobots*, actuated and controlled by a DC voltage signal from a substrate [2], [3]; *light-driven nanocars*, synthetic molecules actuated by a specific wavelength of light [4], *MagMite* microrobots

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

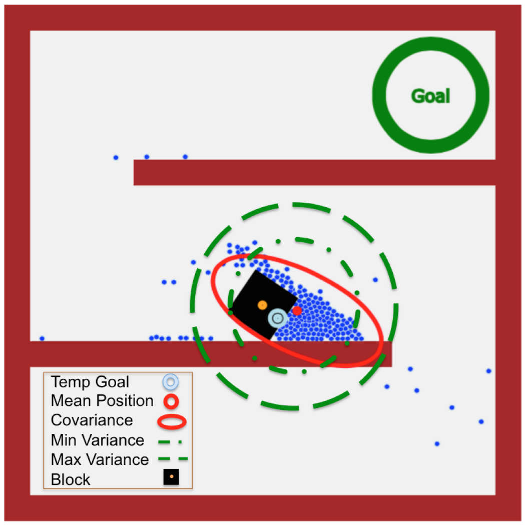

Fig. 1. A swarm of robots, all controlled by a uniform force field, can be effectively controlled by a hybrid controller that knows only the first and second moments of the robot distribution. Here a swarm of simple robots (blue discs) pushes a black block toward the goal.

with different resonant frequencies controlled by a global magnetic field [5]; and magnetically controlled nanoscale helical screws [6], [7]. Large numbers of robots can be constructed, but 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: while one robot is controlled, the rest are ignored. Making progress in targeted therapy and swarm manipulation requires the coordinated control of large robot populations.

B. Human user studies with large swarms

There is currently no comprehensive understanding of user interfaces for controlling multi-robot systems with massive populations [8]. Our previous work with over a hundred hardware robots and thousands of simulated robots [9] 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. To gather better data, we designed *SwarmControl.net*, a large-scale online game to test how humans interact with large swarms [10]. Our goal was to test several scenarios involving large-scale human-swarm interaction, and to do so with a statistically-significant sample size. These experiments showed that numerous simple robots responding to global control inputs are directly controllable by a human operator without special training, and that the visual feedback of the swarm state should be very simple in order to increase task performance. All code [11], and experimental results were posted online. The current paper presents motion primitives and an automatic controller that solves one of the games from SwarmControl.net.

C. Block-pushing and Compliant Manipulation

Unlike *caging* manipulation, where robots form a rigid arrangement around an object [12], [13], 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. [14], 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 [15], [16].

Our n-robot system with 2 control inputs and $4n$ states is inherently under-actuated, and superficially bears resemblance to compliant, under-actuated manipulators [17], [18]. Like these manipulators, the swarm conforms to the object to be manipulated. However our swarm lacks the restoring force provided by flexures in [17] and the silicone in [18]. Our swarm tends to disperse itself, so we require artificial forces, such as the variance control primitives in Section III-D, to regroup the swarm.

III. Theory

A. Models

Consider holonomic robots that move in the 2D plane. We want to control position and velocity of the robots. First, assume a noiseless system containing one robot with mass m. Our inputs are global forces $[u_x, u_y]$. 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:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= A\mathbf{x}(t) + B\mathbf{u}(t) \\ \mathbf{y}(t) &= C\mathbf{x}(t) + D\mathbf{u}(t)\end{aligned} \tag{1}$$

and our state space representation as:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4\end{bmatrix} = \begin{bmatrix}0&1&0&0\\0&0&0&0\\0&0&0&1\\0&0&0&0\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\\x_4\end{bmatrix} + \begin{bmatrix}0&0\\ \frac{1}{m}&0\\0&0\\0&\frac{1}{m}\end{bmatrix}[u_x, u_y] \tag{2}$$

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

$$\mathcal{C} = [B, AB, A^2B, ..., A^{n-1}B]. \tag{3}$$

$$\text{Here } \mathcal{C} = \left[\begin{array}{cc|cc|cc|cc}0&0&\frac{1}{m}&0&0&0&0&0\\ \frac{1}{m}&0&0&0&0&0&0&0\\ 0&0&0&\frac{1}{m}&0&0&0&0\\ 0&\frac{1}{m}&0&0&0&0&0&0\end{array}\right], \tag{4}$$

and thus all four states are controllable.

B. Independent control with multiple robots is impossible

A single robot is fully controllable, but what happens with n robots? For holonomic robots, 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 robots to be controlled in the x axis, there are $2n$ states, n positions and n velocities:

$$[x_1, x_2, \ldots, x_{2n-1}, x_{2n}]^\intercal = [p_{x,1}, v_{x,1}, \ldots, p_{x,n}, v_{x,n}]^\intercal .$$

Our state-space representation is:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \vdots\\ \dot{x}_{2n-1}\\ \dot{x}_{2n}\end{bmatrix} = \begin{bmatrix}0&1&\ldots&0&0\\0&0&\ldots&0&0\\ \vdots&\vdots&\ddots&\vdots&\vdots\\0&0&\ldots&0&1\\0&0&\ldots&0&0\end{bmatrix}\begin{bmatrix}x_1\\x_2\\ \vdots\\x_{2n-1}\\x_{2n}\end{bmatrix} + \begin{bmatrix}0\\1\\ \vdots\\0\\1\end{bmatrix}u_x \tag{5}$$

However, just as with one robot, we can only control two states because $\mathcal{C}$ has rank two:

$$\mathcal{C} = \left[\begin{array}{c|c|c|c|c}0&1&0&0&\\1&0&0&0&\\ \vdots&\vdots&\vdots&\vdots&,\ldots\\0&1&0&0&\\1&0&0&0&\end{array}\right] \tag{6}$$

C. Controlling Mean Position

This means *any* number of robots controlled by a global command have just two controllable states in each axis. We can not control the position of all the robots, but what states are controllable? To answer this question we create the following reduced order system that represents the average x position and x velocity of the n robots:

$$\begin{aligned}\begin{bmatrix}\dot{\bar{x}}_1\\ \dot{\bar{x}}_2\end{bmatrix} &= \frac{1}{n}\begin{bmatrix}0&1&\ldots&0&1\\0&0&\ldots&0&0\end{bmatrix}\begin{bmatrix}x_{1,1}\\x_{2,1}\\ \vdots\\x_{1,n}\\x_{2,n}\end{bmatrix}\\ &+ \frac{1}{n}\begin{bmatrix}0&0&\ldots&0&0\\0&1&\ldots&0&1\end{bmatrix}\begin{bmatrix}0\\1\\ \vdots\\0\\1\end{bmatrix}u_x\end{aligned} \tag{7}$$

Thus:

$$\begin{bmatrix}\dot{\bar{x}}_1\\ \dot{\bar{x}}_2\end{bmatrix} = \begin{bmatrix}0&1\\0&0\end{bmatrix}\begin{bmatrix}\bar{x}_1\\ \bar{x}_2\end{bmatrix} + \begin{bmatrix}0\\1\end{bmatrix}u_x \tag{8}$$

We again analyze $\mathcal{C}$:

$$\mathcal{C} = \left[\begin{array}{c|c}0&1\\1&0\end{array}\right] \tag{9}$$

This matrix has rank two, and thus the average position and average velocity are controllable.

Due to symmetry, only the mean position and mean velocity are controllable. However, there are several techniques for breaking symmetry, for example by allowing independent noise sources [19], or by using obstacles [9].

D. *Controlling the variance of many robots*

The variance, σ_x^2, σ_y^2, of the swarm's position is computed:

$$\overline{x}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} x_{1,i}, \qquad \sigma_x^2(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}(x_{1,i}-\overline{x}_1)^2,$$
$$\overline{y}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} x_{3,i}, \qquad \sigma_y^2(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}(x_{3,i}-\overline{y})^2. \quad (10)$$

Controlling the variance requires being able to increase and decrease the variance. We will list a sufficient condition for each. Both conditions are readily found at the micro and nanoscale. Real systems, especially at the micro scale, are affected by unmodelled dynamics. These unmodelled dynamics are dominated by Brownian noise. To model this (1) must be modified as follows:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= A\mathbf{x}(t) + B\mathbf{u}(t) + W\varepsilon(t) \quad (11)\\ \mathbf{y}(t) &= C\mathbf{x}(t) + D\mathbf{u}(t)\end{aligned}$$

where $W\varepsilon(t)$ is a random perturbation produced by Brownian noise. Given a large free workspace, a *Brownian noise* process increases the variance linearly with time.

$$\dot{\sigma_x^2}(\mathbf{x}(t), \mathbf{u}(t) = 0) = W\varepsilon \quad (12)$$

If robots 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,

$$[\sigma_x^2, \sigma_y^2] = \frac{1}{12}[(\ell_x - 2r)^2, (\ell_y - 2r)^2]. \quad (13)$$

A flat obstacle can be used to decrease variance. Pushing a group of dispersed robots against a flat obstacle will decrease their variance until the minimum-variance (maximum density) packing is reached. For large n, Graham and Sloan showed that the minimum-variance packing $\sigma^2_{optimal}(n, r)$ for n circles with radius r is $\approx \frac{\sqrt{3}}{\pi}(nr)^2 \approx 0.55(nr)^2$ [20].

We will prove the origin is globally asymptotically stabilizable by using a control-Lyapunov function [21]. A suitable Lyapunov function is squared variance error:

$$\begin{aligned}V(t,\mathbf{x}) &= \frac{1}{2}(\sigma^2(\mathbf{x}) - \sigma^2_{goal})^2\\ \dot{V}(t,\mathbf{x}) &= (\sigma^2(\mathbf{x}) - \sigma^2_{goal})\dot{\sigma^2}(\mathbf{x}) \quad (14)\end{aligned}$$

We note here that $V(t,\mathbf{x})$ is positive definite and radially unbounded, and $V(t,\mathbf{x}) \equiv 0$ only at $\sigma^2(\mathbf{x}) = \sigma^2_{goal}$. To make $\dot{V}(t,\mathbf{x})$ negative semi-definite, we choose

$$u(t) = \begin{cases}\text{move to wall} & \text{if } \sigma^2(\mathbf{x}) > \sigma^2_{goal}\\ \text{move from wall} & \text{if } \sigma^2(\mathbf{x}) \le \sigma^2_{goal}.\end{cases} \quad (15)$$

For such a $u(t)$,

$$\dot{\sigma^2}(\mathbf{x}) = \begin{cases}\text{negative} & \text{if } \sigma^2(\mathbf{x}) > \max(\sigma^2_{goal}, \sigma^2_{optimal}(n,r))\\ W\epsilon & \text{if } \sigma^2(\mathbf{x}) \le \sigma^2_{goal},\end{cases} \quad (16)$$

and thus $\dot{V}(t,\mathbf{x})$ is negative definite and the variance is globally asymptotically stabilizable.

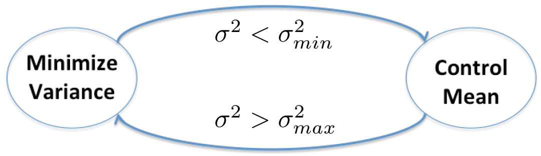

Fig. 2. Two states for controlling the mean and variance of a robot swarm.

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

The mean and variance of the swarm cannot be controlled simultaneously, however if the dispersion due to Brownian motion is much less than the maximum controlled speed, we can adopt a hybrid, hysteresis-based controller to regulate the mean and variance shown in Alg. 1. Such a controller normally controls the mean position according to (18), but switches to minimizing variance if the variance exceeds some σ^2_{max}. The variance is lowered to less than σ^2_{min}, and the system returns to controlling the mean position. This is a standard technique for dealing with control objectives that evolve at different rates [22], [23], and the hysteresis avoids rapid switching between control modes. The process is illustrated in Fig. 2.

Algorithm 1 Hybrid mean and variance control

```
Require: Knowledge of swarm mean $[\bar{x}, \bar{y}]$, the locations
    of the rectangular boundary $\{x_{min}, x_{max}, y_{min}, y_{max}\}$,
    and the target mean position $[x_{target}, y_{target}]$.
 1: $flag_x \leftarrow$ false, $flag_y \leftarrow$ false
 2: $x_{goal} \leftarrow x_{target}$, $y_{goal} \leftarrow y_{target}$
 3: loop
 4:     Compute $\sigma_x^2, \sigma_y^2$
 5:     if $\sigma_x^2 > \sigma^2_{max}$ then
 6:         $x_{goal} \leftarrow x_{min}$
 7:         $flag_x \leftarrow$ true
 8:     else if $flag_x$ and $\sigma_x^2 < \sigma^2_{min}$
 9:         $x_{goal} \leftarrow x_{target}$
10:         $flag_x \leftarrow$ false
11:     end if
12:     if $\sigma_y^2 > \sigma^2_{max}$ then
13:         $y_{goal} \leftarrow y_{min}$
14:         $flag_y \leftarrow$ true
15:     else if $flag_y$ and $\sigma_y^2 < \sigma^2_{min}$
16:         $y_{goal} \leftarrow y_{target}$
17:         $flag_y \leftarrow$ false
18:     end if
19:     Apply (18) to move toward $[x_{goal}, y_{goal}]$
20: end loop
```

A key challenge is to select proper values for σ^2_{min} and σ^2_{max}. The optimal packing variance is $\sigma^2_{optimal}(n, r) = nr^2\frac{\sqrt{3}}{2\pi}$. The random packings generated by pushing our robots into corners are suboptimal, so we choose the conservative values shown in Fig. 3:

$$\begin{aligned}\sigma^2_{min} &= 2.5r + \sigma^2_{optimal}(n,r)\\ \sigma^2_{max} &= 15r + \sigma^2_{optimal}(n,r). \quad (17)\end{aligned}$$

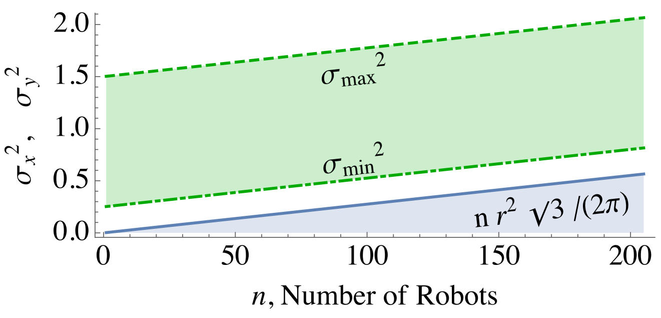

Fig. 3. The switching conditions for variance control are set as a function of n, and designed to be larger than the optimal packing density. The above plot uses robot radius $r = 1/10$.

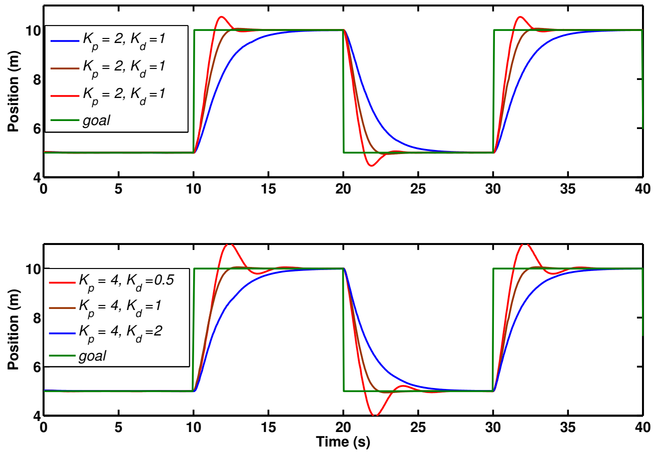

Fig. 4. Tuning proportional (K_p, top) and derivative (K_d, bottom) gain values in (18) improves performance with n= 100 robots.

IV. Simulation

Our simulations use a Javascript port of Box2D, a popular 2D physics engine with support for rigid-body dynamics and fixed-timestep simulation.

A. Controlling the mean position

We control mean position with a PD controller that uses the mean position and mean velocity. Our control input is the global force applied to each robot:

$$u_x = K_p(x_{goal} - \bar{x}) + K_d(0 - \bar{v}_x)$$
$$u_y = K_p(y_{goal} - \bar{y}) + K_d(0 - \bar{v}_y) \qquad (18)$$

here K_p is the proportional gain, and K_d is the derivative gain. We performed a parameter sweep to identify the best values. Representative experiments are shown in Fig. 4. 100 robots were used and the maximum speed was 3 meters per second. As shown in Fig. 4, we can achieve an overshoot of 1% and a rise time of 1.52 s with $K_p = 4$, and $K_d = 1$.

B. Controlling the variance

For variance control we use the control law discussed in Section III-D. Moving away from the wall and waiting is sufficient to increase variance because Brownian noise naturally disperses the swarm in such a way that the variance increases linearly [24]. If faster dispersion is needed, the swarm can be pushed through obstacles such as a diffraction grating or Pachinko board [9].

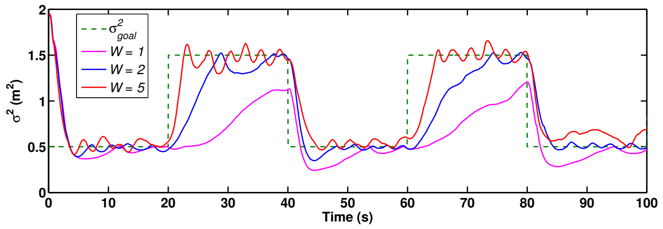

Fig. 5. Increased noise results in more responsive variance control because stronger Brownian noise causes a faster increase of variance.

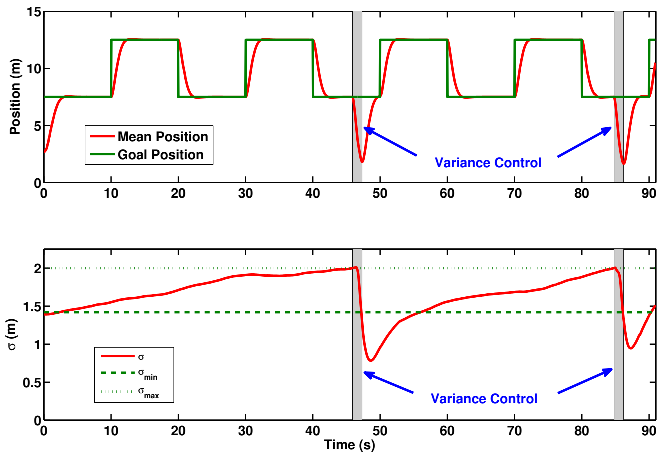

Fig. 6. Simulation result with 100 robots under hybrid control Alg. 1, which controls both the mean position (top) and variance (bottom). For ease of analysis, only x position and variance are shown.

The variance control law to regulate the variance to σ_{ref}^2 has three gains:

$$u_x = K_p(x_{goal}(\sigma_{ref}^2) - \bar{x}) - K_d\bar{v}_x + K_i(\sigma_{ref}^2 - \sigma_x^2)$$
$$u_y = K_p(y_{goal}(\sigma_{ref}^2) - \bar{y}) - K_d\bar{v}_y + K_i(\sigma_{ref}^2 - \sigma_y^2). \qquad (19)$$

In a slight abuse of notation we call the gain scaling the variance error K_i because the variance integrates over time. Eq. (19) assumes the nearest wall is to the left of the robot at $x = 0$, and chooses a reference goal position that in steady-state would have the correct variance according to (13):

$$x_{goal}(\sigma_{ref}^2) = r + \sqrt{3\sigma_{ref}^2} \qquad (20)$$

If another wall is closer, the signs of $[K_p, K_i]$ are inverted, and the location x_{goal} is translated. Results are shown in Fig. 5, with $K_{p,i,d} = [4, 1, 1]$.

C. Hybrid Control of mean and variance

Fig. 6 shows a simulation run of the hybrid controller in Alg. 1 with 100 robots in a square workspace containing no internal obstacles.

V. Block-Pushing Results

This section analyzes a *block-pushing* task attempted by both our hybrid hysteresis-based controller and by human users. All experiments used the 2D physics engine box2D [25], running in a Chrome web browser on a 2.6 GHz Macbook. All code is available at [26].

A. Human-Controlled Block-Pushing

In previous work over 1000 human users completed an online version of this task using varying levels of feedback. The original experiment explored manipulation with varying amounts of sensing information: **full-state** sensing showed the position of all robots; **convex-hull** drew a convex hull around the outermost robots; **mean** displayed the average position of the population; and **mean + variance** added a confidence ellipse. Fig. 7 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. We hypothesized a steady decay in performance as the amount of visual feedback decreased.

To our surprise, the results indicated the opposite: players with just the mean completed the task faster than those with full-state feedback. As Fig. 8 shows, the levels of feedback arranged by increasing completion time are [mean, mean + variance, full-state, convex-hull]. Interviews with beta-testers suggests that tracking 100 robots was overwhelming—similar to schooling phenomenons that confuse predators—while working with just the mean + variance was like using a “spongy” manipulator. Convex-hull feedback was confusing and irritating because a single robot left behind an obstacle would distort the entire hull, obscuring the information about the majority of the swarm.

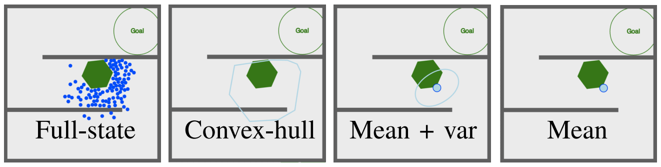

Fig. 7. Screenshots from a block-pushing task with human users. This experiment challenged players to quickly steer 100 robots (blue discs) to push an object (green hexagon) into a goal region.

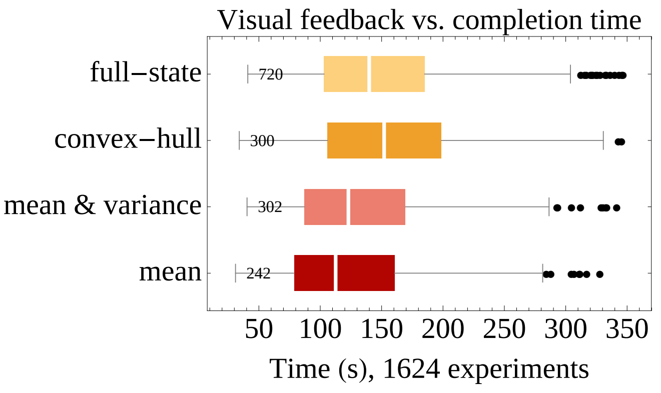

Fig. 8. Completion-time results for the four levels of visual feedback shown in Fig. 7.

Algorithm 2 Block-pushing controller for a robotic swarm.

```
Require: Knowledge of swarm mean $[\bar{x},\bar{y}]$, the moveable block's center of mass $\mathbf{b}$, a map of the environment, and the locations of all convex corners $\mathbf{C}$
Require: Robot distribution is unimodal
Require: Obstacle-free, straight-line path from swarm to moveable block
 1: Compute $\mathbf{M}$, the distance to goal, with breadth-first search
 2: Compute the gradient, $\nabla\mathbf{M}$
 3: $\mathbf{C} \leftarrow$ sort($\mathbf{C}$) according to $-\mathbf{M}$
 4: while b is not in goal region do
 5:     $\sigma^2 \leftarrow \max(\sigma_x, \sigma_y)$
 6:     if $\sigma^2 > \sigma^2_{max}$ then
 7:         while $\sigma^2 > \sigma^2_{min}$ do
 8:             $\mathbf{c}_i \leftarrow$ the nearest corner in $\mathbf{C}$ to $[\bar{x},\bar{y}]$
 9:             $[x_{goal}, y_{goal}] \leftarrow \mathbf{c}_i$
10:             if $\mathbf{M}(\mathbf{b}) > \mathbf{M}(\mathbf{c}_i)$ then
11:                 $[x_{goal}, y_{goal}] \leftarrow \mathbf{c}_{i-1}$
12:                 Apply (18) to move toward $[x_{goal}, y_{goal}]$
13:             end if
14:         end while
15:     else
16:         if distance($\mathbf{b}, [x_{goal}, y_{goal}]$) $> k_1 r_b$ then
17:             $r_p \leftarrow k_2 r_b$                    ▷ guarded move
18:         else
19:             $r_p \leftarrow k_3 r_b$                    ▷ pushing move
20:         end if
21:         $[x_{goal}, y_{goal}] \leftarrow \mathbf{b} - r_p \nabla\mathbf{M}(\mathbf{b})$
22:     end if
23:     Apply (18) to move toward $[x_{goal}, y_{goal}]$
24: end while
```

B. Automated Block-Pushing

Fig. 9 shows snapshots during an execution of this algorithm. To solve this block-pushing task, we discretized the environment. On this discretized grid we used breadth-first search to determine $\mathbf{M}$, the shortest distance from any grid cell to the goal, and generated a gradient map $\nabla\mathbf{M}$ toward the goal as shown in Fig. 10. The block's center of mass is at $\mathbf{b}$ and has radius r_b. Three constants are needed, where $k_1 > k_2 > 1$ and $1 > k_2 > 0$. All experiments used $[k_1, k_2, k_3] = [2.5, 1.5, 0.1]$. The robots were directed to assemble behind the block at $\mathbf{b} - k_2 r_b \nabla\mathbf{M}(\mathbf{b})$, then move to $\mathbf{b} - k_3 r_b \nabla\mathbf{M}(\mathbf{b})$ to push the block toward the goal location. We use the hybrid hysteresis-based controller in Alg. 1 to track the desired position, while maintaining sufficient robot density to move a block by switching to minimize variance whenever variance exceeds a set limit. The minimize variance control law (19) is slightly modified to choose the nearest corner further from the goal than $\mathbf{b}$ with an obstacle-free straight-line path to $\mathbf{b}$. The control algorithm for block-pushing is listed in Alg. 2. Experimental results are summarized in Fig. 11. Although larger populations of robots can apply more force, minimizing the variance requires more

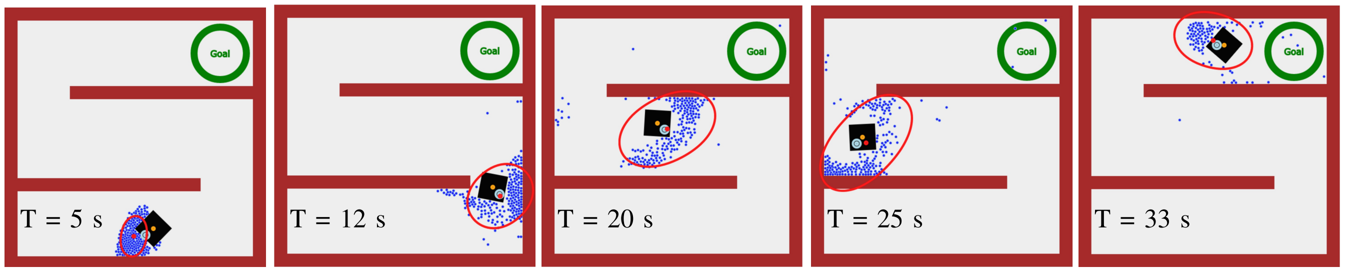

Fig. 9. Snapshots showing the block-pushing experiment with 200 robots under automatic control. See the video attachment for an animation.

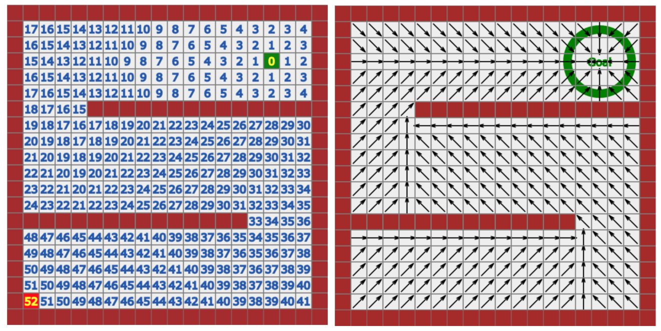

Fig. 10. The BFS algorithm finds the shortest path for the moveable block (left), which is used to compute gradient vectors (right).

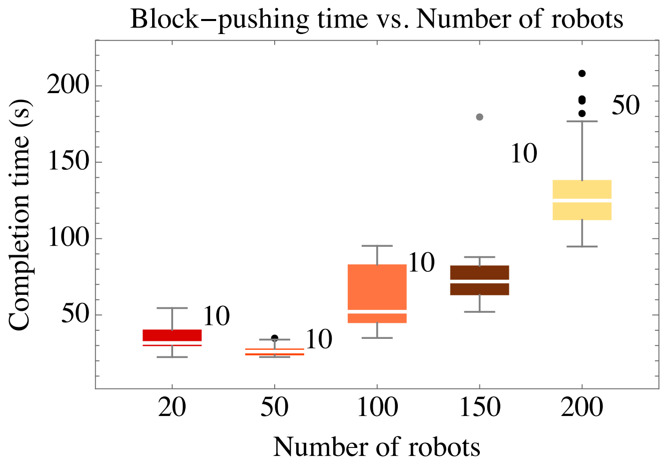

Fig. 11. Completion-time results using the automatic controller from Alg. 2 for different numbers of robots. Each bar is labelled with the number of trials.

time with larger populations and dominates task completion time.

Algorithm 2 is an imperfect solution and has a failure mode if the robot swarm becomes multi-modal with modes separated by an obstacle, as shown in Fig. 12. In this case, moving toward a corner will never reduce the variance below σ^2_{min}.

The first challenge is to identify when the distribution has become multi-modal. Measuring just the mean and variance is insufficient to determine if a distribution is no longer unimodal, but if the swarm is being directed to a corner, and the variance does not reduce below σ^2_{min}, the swarm has become separated. In this case, we must either manipulate with a partial swarm, or run a gathering algorithm. For the ‘S’-shaped workspace in this study, an open-loop input that

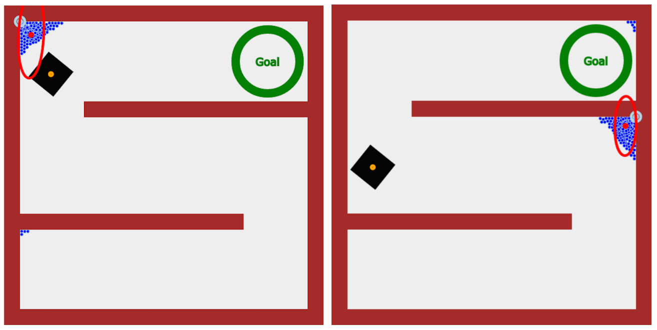

Fig. 12. Algorithm 2 fails when some robots are separated by the maze and the swarm can not achieve $\sigma^2 < \sigma^2_{min}$. These failures occured during 14% of trials.

commands the swarm to move in succession {LEFT, DOWN, RIGHT, DOWN} will move the swarm to the bottom right corner. This is not true for all obstacle fields. In a ‘T’-shaped workspace, it is not possible to find an open-loop input that will move the entire swarm to the bottom of the ‘T’.

Using only the mean and variance may be overly restrictive. Many heuristics using high-order moments have been developed to test if a distribution is multimodal [27]. Often the sensor data itself, though it may not resolve individual robots, will indicate multi-modality. For instance CCD images reveal clusters of bacteria, and MRI scans show agglomerations of particles [28]. This data can be fitted with K-means or expectation maximization algorithms, and manipulation could be performed with the nearest swarm of sufficient size.

VI. Conclusion and Future Work

Inspired by large-scale human experiments with swarms of robots under global control, this paper investigated controllers that use only the mean and variance of a robot swarm. We proved that the mean position is controllable, and provided conditions under which variance is controllable. We derived automatic controllers for each and a hysteresis-based switching control that controls the mean and variance of a robot swarm. We employed these controllers as primitives for a block-pushing task.

Future work should implement these controllers on a robot swarm and decrease completion time by avoiding counter-productive contact with the block while the swarm is lowering its variance. We have also assumed the swarm is unimodal and has a straight-line path to the moveable block. Relaxing these assumptions requires solving the *gathering* *problem*. The gathering problem for a swarm with uniform inputs is largely unexplored, and must be examined probabilistically for nontrivial environments.

References

[1] 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," *Journal of Micro-Bio Robotics*, vol. 9, no. 1-2, pp. 23–28, 2014.

[2] B. Donald, C. Levey, C. McGray, I. Paprotny, and D. Rus, "An untethered, electrostatic, globally controllable MEMS micro-robot," *J. of MEMS*, vol. 15, no. 1, pp. 1–15, Feb. 2006.

[3] B. Donald, C. Levey, and I. Paprotny, "Planar microassembly by parallel actuation of MEMS microrobots," *J. of MEMS*, vol. 17, no. 4, pp. 789–808, Aug. 2008.

[4] P.-T. Chiang, J. Mielke, J. Godoy, J. M. Guerrero, L. B. Alemany, C. J. Villagómez, A. Saywell, L. Grill, and J. M. Tour, "Toward a light-driven motorized nanocar: Synthesis and initial imaging of single molecules," *ACS Nano*, vol. 6, no. 1, pp. 592–597, Feb. 2011.

[5] D. Frutiger, B. Kratochvil, K. Vollmers, and B. J. Nelson, "Magmites - wireless resonant magnetic microrobots," in *IEEE Int. Conf. Rob. Aut.*, Pasadena, CA, May 2008.

[6] S. Tottori, L. Zhang, F. Qiu, K. Krawczyk, A. Franco-Obregón, and B. J. Nelson, "Magnetic helical micromachines: Fabrication, controlled swimming, and cargo transport," *Advanced Materials*, vol. 24, no. 811, 2012.

[7] K. E. Peyer, L. Zhang, and B. J. Nelson, "Bio-inspired magnetic swimming microrobots for biomedical applications," *Nanoscale*, 2013.

[8] S. Nunnally, P. Walker, A. Kolling, N. Chakraborty, M. Lewis, K. Sycara, and M. Goodrich, "Human influence of robotic swarms with bandwidth and localization issues," in *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*, Oct 2012, pp. 333–338.

[9] A. 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 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2013, pp. 520–527.

[10] A. Becker, C. Ertel, and J. McLurkin, "Crowdsourcing swarm manipulation experiments: A massive online user study with large swarms of simple robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014. [Online]. Available: https://arxiv.org/submit/913241/view

[11] C. Ertel and A. Becker. (2013, Sep.) Swarmcontrol git repository. [Online]. Available: https://github.com/crertel/swarmmanipulate.git

[12] 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.

[13] J. Fink, N. Michael, and V. Kumar, "Composition of vector fields for multi-robot manipulation via caging," in *Robotics Science and Systems*, Atlanta, GA, Jun. 2007.

[14] K. Lynch, "Locally controllable manipulation by stable pushing," *IEEE Trans. Robot. Autom.*, vol. 15, no. 2, pp. 318–327, Apr. 1999.

[15] M. D. Armani, S. V. Chaudhary, R. Probst, and B. Shapiro, "Using feedback control of microflows to independently steer multiple particles," *Journal of Microelectromechanical systems*, vol. 15, no. 4, Aug. 2006.

[16] A. Becker, R. Sandheinrich, and T. Bretl, "Automated manipulation of spherical objects in three dimensions using a gimbaled air jet," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, oct. 2009, pp. 781 –786. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5354427

[17] L. U. Odhner, L. P. Jentoft, M. R. Claffee, N. Corson, Y. Tenzer, R. R. Ma, M. Buehler, R. Kohout, R. D. Howe, and A. M. Dollar, "A compliant, underactuated hand for robust manipulation," *The International Journal of Robotics Research*, vol. 33, no. 5, pp. 736–752, 2014.

[18] R. Deimel and O. Brock, "A novel type of compliant, underactuated robotic hand for dexterous grasping," *Robotics: Science and Systems, Berkeley, CA*, pp. 1687–1692, 2014.

[19] A. Becker, C. Onyuksel, T. Bretl, and J. McLurkin, "Controlling many differential-drive robots with uniform control inputs," *The international journal of Robotics Research*, vol. 33, no. 13, pp. 1626–1644, 2014.

[20] R. L. Graham and N. J. Sloane, "Penny-packing and two-dimensional codes," *Discrete & Computational Geometry*, vol. 5, no. 1, pp. 1–11, 1990.

[21] A. M. Lyapunov, translated and edited by A.T. Fuller, *The General Problem of the Stability of Motion*. London: Tayor & Francis, 1992.

[22] S. Sadraddini and C. Belta, "Swarm manipulation with a uniform magnetic field," in *unpublished*, 2015.

[23] M. Kloetzer and C. Belta, "Temporal logic planning and control of robotic swarms by hierarchical abstractions," *Robotics, IEEE Transactions on*, vol. 23, no. 2, pp. 320–330, 2007.

[24] A. Einstein, *Investigations on the Theory of the Brownian Movement*. Courier Corporation, 1956.

[25] E. Catto, "User manual, Box2D: A 2D physics engine for games, http://www.box2d.org," 2010. [Online]. Available: http://www.box2d.org

[26] S. Shahrokhi and A. T. Becker, "BlockPushingIROS2015, https://github.com/aabecker/swarmcontrolsandbox/blob/master/examplecontrollers/blockpushingiros2015.html," Jul. 2015.

[27] J. Haldane, "Simple tests for bimodality and bitangentiality," *Annals of eugenics*, vol. 16, no. 1, pp. 359–364, 1951.

[28] M. Stuber, W. D. Gilson, M. Schär, D. A. Kedziorek, L. V. Hofmann, S. Shah, E.-J. Vonken, J. W. Bulte, and D. L. Kraitchman, "Positive contrast visualization of iron oxide-labeled stem cells using inversion-recovery with on-resonant water suppression (iron)," *Magnetic Resonance in Medicine*, vol. 58, no. 5, pp. 1072–1077, 2007.