Contents

Steering a Swarm Using Global Inputs and Swarm Statistics

Shiva Shahrokhi, Chris Ertel, Mable Wan,
Lillian Lin, Aaron T. Becker*

August 15, 2016

Abstract

Micro- and nanorobotics have 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 to generating controlled motion. We want to use large swarms of robots to perform manipulation tasks; unfortunately, human-swarm interaction studies as conducted today are limited in sample size, are difficult to reproduce, and are prone to hardware failures. We present an alternative.

This paper first examines the perils, pitfalls, and possibilities we discovered by launching SwarmControl.net, an online game where players steer swarms of up to 500 robots to complete manipulation challenges. We record statistics from thousands of players, and use the game to explore aspects of large-population robot control. We present the game framework as a new, open-source tool for large-scale user experiments. One surprising result was that humans completed an object manipulation 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 next investigates controllers that use only the mean and variance of a robot swarm. We prove that the mean position is controllable, then provide conditions under which variance is 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 an object manipulation task and implement all the automatic controllers on 100 kilobots controlled by the direction of a global light source.

*Authors are with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX, 77004 USA, sshahrokhi2@uh.edu, atbecker@uh.edu

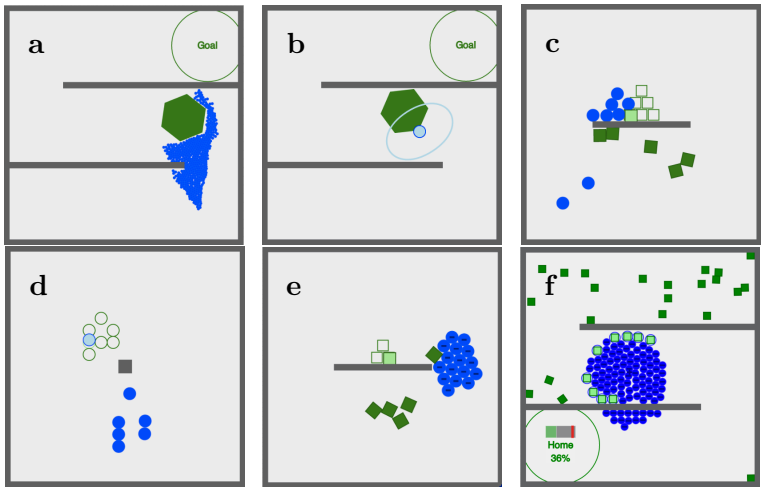


Figure 1: Screenshots from our online experiments controlling multi-robot systems with limited, global control. **(a)** Varying the number of robots from 1-500 **(b)** Comparing 4 levels of visual feedback **(c)** Varying noise from 0 to 200% of control authority **(d)** Controlling the position of 1 to 10 robots **(e)** Comparing 3 control architectures to assemble **(f)** Comparing 3 control architectures to forage.

1 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 Peyer et al. (2013); Shirai et al. (2005); Chiang et al. (2011). These activities require robots that behave intelligently. Limited computation and communication rules out autonomous operation or direct control over individual units; instead we must rely on global control signals broadcast to the entire robot population. Additionally, many promising applications will require direct human control, but user interfaces to thousands—or millions—of robots is a daunting human-swarm interaction (HSI) challenge.

Our previous work with over a hundred hardware robots and thousands of simulated robots Becker et al. (2013) 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. There is currently no comprehensive understanding of user interfaces for controlling multi-robot systems with massive populations. This paper presents a tool for investigating HSI methods through statistically significant numbers of experiments.

Our goal was to test several scenarios involving large-scale human-swarm interaction (HSI), and to do so with a statistically-significant sample size. Towards this end, we created SwarmControl.net, an open-source, online testing platform suitable for inexpensive deployment and data collection on a scale not yet seen in swarm robotics research. Screenshots from this platform are shown in Fig. 1. All code and experimental results are posted online at Ertel et al. (2016).

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

Often robots 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 Martel et al. (2014), 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, inspired by our online experiments, that require only mean and variance measurements of the robot's positions. These controllers are used as primitives to perform the object manipulation task illustrated in Fig. 2.

Our paper is organized as follows. After a discussion of related work in §2, we describe our experimental methods for an online human-user experiment in §3. We report the results of our online experiments in §3.2. Inspired by these results, we prove that the mean and variance of a robot swarm are controllable in §4, and present automatic controllers in §5. We use these controllers as primitives and present a framework for manipulating an object through a maze in §6. We conclude with implementations of these controllers in our hardware robots and use them to complete an object manipulation task with 100+ kilobots in §7.

This paper is the synthesis of two preliminary conference papers. Becker et al. (2014a), covered the first three months of SwarmControl.net experiments, and Shahrokhi and Becker (2015), presented simulations of object manipulation. This paper presents the final results from SwarmControl.net. All hardware validation experiments are new.

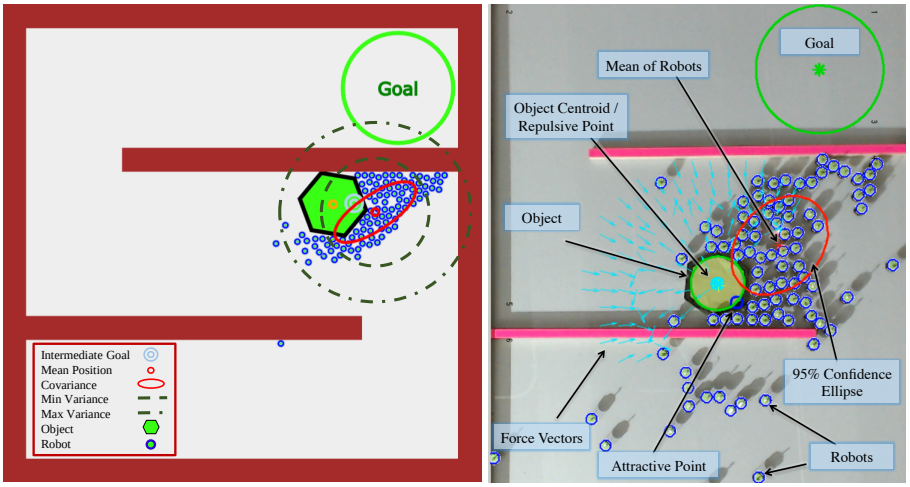


Figure 2: A swarm of robots, all controlled by a uniform force field, can be effectively controlled by a hybrid controller that knows only the mean and variance of the robot distribution. Here a swarm of simple robots (blue discs) pushes a green hexagon toward the goal. Simulation at left, screen capture from hardware experiment at right. See multimedia extension.

2 Related Work

This section describes global control challenges and reviews highlights of human-swarm interaction, block pushing, and compliant manipulation.

2.1 Global control of microrobots

Small robots have been constructed with physical heterogeneity so that they respond differently to a global, broadcast control signal. Examples include *scratch-drive microrobots*, actuated and controlled by a DC voltage signal from a substrate by Donald et al. (2006, 2008); magnetic structures with different cross-sections that could be independently steered by Floyd et al. (2011); Diller et al. (2013); *MagMite* microrobots with different resonant frequencies and a global magnetic field by Frutiger et al. (2008); and magnetically controlled nanoscale helical screws constructed to stop movement at different cutoff frequencies of a global magnetic field by Tottori et al. (2012) and Peyer et al. (2013).

Similarly, our previous work, Becker and Bretl (2012); Becker et al. (2012), focused on exploiting inhomogeneity between robots. These control algorithms theoretically apply to any number of robots—even robotic

continuums—but in practice process noise cancels the differentiating effects of inhomogeneity for more than tens of robots. We desire control algorithms that extend to many thousands of robots.

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

Control—global inputs: Many micro- and nanorobotic systems, see Tottori et al. (2012); Shirai et al. (2005); Chiang et al. (2011); Donald et al. (2006, 2008); Takahashi et al. (2006); Floyd et al. (2011); Diller et al. (2013); Frutiger et al. (2008); Peyer et al. (2013) rely on global inputs, where each robot receives an exact copy of the control signal. Our experiments follow this global model.

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

Computation—calculating the control law: In our previous work the controllers required at best a summation over all the robot states, see Becker et al. (2012) and at worst a matrix inversion, see Becker and Bretl (2012). 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.

2.2 Human-swarm interaction

Olsen Jr. and Wood (2004) studied human *fanout*, the number of robots a single human user could control. They postulated that the optimal number of robots was approximately the autonomous time divided by the interaction time required by each robot. Their sample problem involved a multi-robot search task, where users could assign goals to robots. Their user interaction

studies with simulated planar robots indicated a *fanout plateau* of about 8 robots, with diminishing returns for more robots. They hypothesized the location of this plateau is highly dependent on the underlying task. Indeed, our paper indicates there are tasks without plateaus. Their research investigated robots with 3 levels of autonomy. We use robots without autonomy, corresponding with their first-level robots.

Squire et al. (2006) designed experiments showing user-interface design had a high impact on the task effectiveness and the number of robots that could be controlled simultaneously in a multi-robot task.

Several user studies compare methods for controlling large swarms of simulated robots, for example Bashyal and Venayagamoorthy (2008); Kolling et al. (2012); de la Croix and Egerstedt (2012). 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 Bashyal and Venayagamoorthy (2008), de la Croix and Egerstedt (2012), and Kolling et al. (2012) 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.

2.3 Block pushing and compliant manipulation

Unlike *caging* manipulation, where robots form a rigid arrangement around an object, as in Sudsang et al. (2002); Fink et al. (2008), 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. Lynch (1999), 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 Armani et al. (2006) and Becker et al. (2009).

Our n-robot system with 2 control inputs and $4n$ states is inherently under-actuated, and superficially bears resemblance to compliant, under-actuated manipulators. Our swarms conform to the object to be manipulated. However our swarms lack the restoring force provided by flexures in Odhner et al. (2014) or silicone in Deimel and Brock (2014). Our swarms tend to disperse, so we require artificial forces, such as the variance control primitives in §4.3, to regroup the swarms.

3 Online experiment

We developed a flexible testing framework for online human-swarm interaction studies. Over 5,000 humans performed over 20,000 swarm-robotics experiments with this framework, logging almost 700 hours of experiments.

3.1 Methods

Our web server generates a unique identifier for each participant and sends it along with the landing page to the participant. This identifier is stored as a browser cookie and is attached to all results the participant generates. A script on the participant's browser runs the experiment and posts the experiment data to the server.

A participant may view all of the experimental data we have gathered; this information is available as either a webpage, a JSON file, or a comma-separated value file.

During the **instantiation** phase, an experiment sets up the web page elements with help text and other information, and creates the obstacles, robots, and workpieces that will be present during the experiment. It will also randomly select which mode to run in, if applicable.

During the **simulation** phase the robots are moved according to user input and given a chance to interact with each other and the environment. The current state of the experiment is then drawn.

In the **evaluation** phase the experiment's completion criteria are applied to the current experiment state: are the robots in the goal zone, are the workpieces in the correct place, and so forth. If the criteria are not met, the experiment repeats the simulation phase; if they are met, then the experiment proceeds to result submission.

In the **submission** phase the experiment results are combined with other user data, such as the browser user agent string, and submitted to the server for collection.

Anonymized human subject data was collected under IRB #14357-01.

3.2 Human-swarm interaction results

We designed six experiments to investigate human control of large robotic swarms for manipulation tasks. Screenshots of each experiment are shown in Fig. 1. Each experiment examined the effects of varying a single parameter: population of robots for manipulation, four levels of visual feedback, different levels of Brownian noise, position control with 1 to 10 robots, and

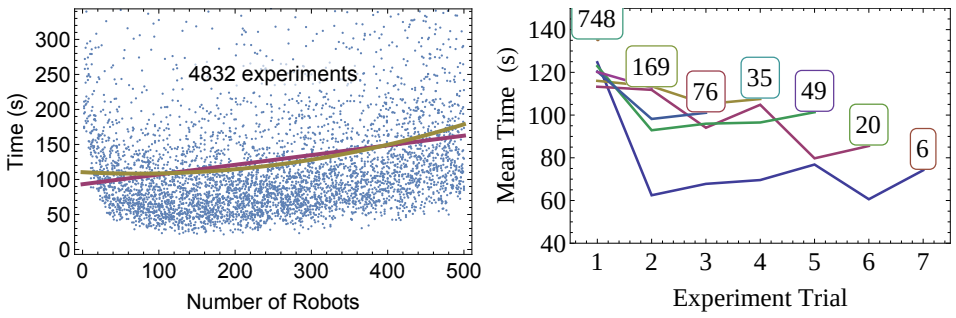


Figure 4: Data from *Varying Number* using robots to push an object through a maze to a goal location. (Left) Data indicates this task has an optimal number of robots, perhaps due to the relative sizes of the robots, obstacles, and object. Best-fit linear and quadratic lines are overlaid for comparison. (Right) Skill improves as players retry tests using data from *Varying Number*.

three control architectures for both an assembly task and a foraging task. The users could choose which experiment to try, and then our architecture randomly assigned a particular parameter value for each trial. We recorded the completion time and the participant ID for each successful trial.

Varying number Transport of goods and materials between points is at the heart of all engineering and construction in real-world systems. This experiment varied from 1 to 500 the population of robots used to transport an object. The total area, maximum robot speed, and total net force the swarm could produce were constant. The robots pushed a large hexagonal object through an S-shaped maze. We hypothesized participants would complete the task faster with more robots. The results, shown in Fig. 4, do not support our hypothesis, indicating a minimum around 130 robots.

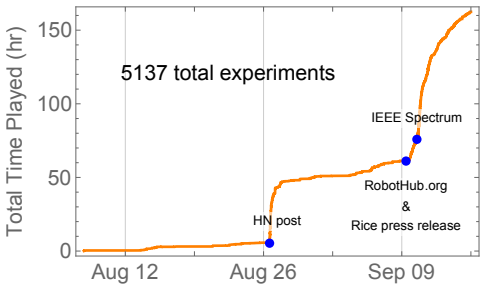


Figure 3: Cumulative time played for completed tests during first two months.

Varying visualization Sensing is expensive, especially on the nanoscale. To see nanocars Chiang et al. (2011) fastened molecules that fluoresce light when activated by a strong light source. Unfortunately, multiple exposures

can destroy these molecules, a process called *photobleaching*. Photobleaching can be minimized by lowering the excitation light intensity, but Cazes (2005) showed this increases the probability of missed detections. This experiment explores manipulation with varying amounts of sensing information: **full-state** sensing provides the most information by showing the position of all robots; **convex-hull** draws a convex hull around the outermost robots; **mean** provides the average position of the population; and **mean + variance** adds a confidence ellipse. Fig. 5 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 according to Har-Peled (2011), the expected number is $O(2n^{1/3})$. Mean requires two, and variance three, data points. Mean and mean + variance are convenient even with millions of robots.

Our hypothesis predicted a steady decay in performance as the amount of visual feedback decreased. Our experiment indicated the opposite: players with just the mean completed the task faster than those with full-state feedback. As Fig. 6.b shows, the levels of feedback arranged by increasing completion time are [mean, mean + variance, full-state, convex-hull]. All experiments lasting over 300s were removed, under the assumption that the user stopped playing. Using ANOVA analysis, we reject 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

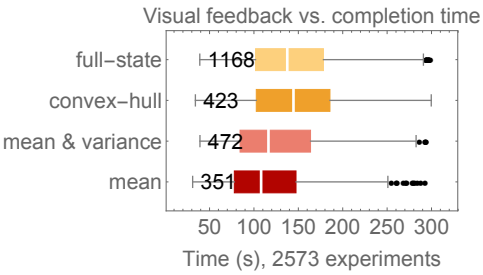


Figure 6: Completion-time results for the four levels of visual feedback shown in Fig. 5. Players performed better with limited feedback.

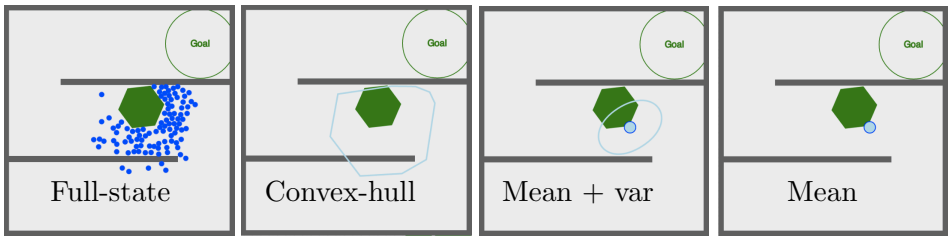


Figure 5: Screenshots from task *Vary Visualization*. This experiment challenges players to quickly steer 100 robots (blue discs) to push an object (green hexagon) into a goal region. We record the completion time and other statistics.

tracking 100 robots is overwhelming—similar to schooling phenomenons that confuse predators—while working with just the mean + variance is like using a "spongy" manipulator. Our beta-testers described convex-hull feedback as confusing and irritating. A single robot left behind an obstacle will stretch the entire hull, obscuring the majority of the swarm.

Varying noise Microrobots are affected by random collisions with molecules. The effect of these collisions is called Brownian motion.

This experiment varied the strength of these disturbances to study how noise affects human control of large swarms. Noise was applied at every timestep as follows:

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

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

We hypothesized 200% noise was the largest a human could be expected to control—at 200% noise, the robots move erratically. Disproving our hypothesis, the results in Fig. 7.a show only a 40% increase in completion time for the maximum noise.

Position control This online experiment examined how completion time scales with the number of robots n. Using a single square obstacle, users arranged $n \in [1, 10]$ robots into a specified goal pattern. The goal pattern

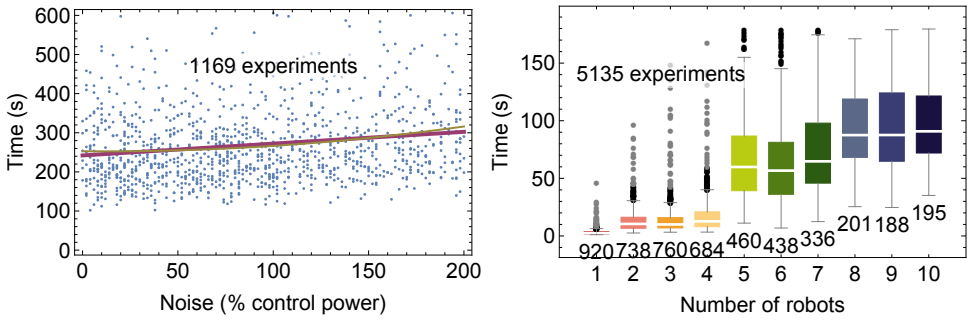


Figure 7: Left: Varying the noise from 0 to 200% of the maximum control input resulted in only a small increase in completion time. Right: Increasing the number of robots resulted in longer completion times. For more than 4 robots the goal pattern contained a void, which may have caused the jump in completion times.

formed a block A with 10 robots, and lesser numbers of robots used a subset of these goal positions. Our hypothesis was that completion time would increase linearly with the number of robots, as with our position control algorithm in Becker et al. (2013). Our results roughly corroborate this, as shown in Fig. 7.b. Though the number of robots presented to game players is uniformly distributed, larger n are more difficult, and the number of successful experiments drops steadily as n increases.

Note there is a bifurcation between n=4 and n=5 robots. For $n \in [1, 4]$ the goal patterns are not hollow, but starting at n=5 they are. A better experiment design would randomly place the goal positions. Initially we tried this, but our beta-testers strongly disliked trying to arrange robots in random patterns.

Varying control: Assembly Ultimately, we want to use swarms of robots to build things. This experiment compared different control architectures modeled after real-world devices.

We compared attractive and repulsive control with the global control used for the other experiments. The attractive and repulsive controllers were loosely modeled after scanning tunneling microscopes (STM), but also apply to magnetic manipulation, e.g. Khalil et al. (2013) and biological models, e.g. Goodrich et al. (2012). STMs can be used to arrange atoms and make small assemblies, as described in Avouris (1995). An STM tip is charged with electrical potential, and used to repel like-charged or to attract differently-charged molecules. In contrast, the global controller uses a uniform field (perhaps formed by parallel lines of differently-charged conductors) to pull molecules in the same direction. The experiment challenged players to assemble a three-block pyramid with a swarm of 16 robots.

The results were conclusive, as shown in Fig. 8.a: attractive control was the fastest, followed by global control, with repulsive control a distant last. The median time using repulsive control was four times longer than with attractive control. Using ANOVA analysis, we reject the null hypothesis that all controllers are equivalent, with p-value 3.37×10^{-32}.

Varying control: Foraging Collecting and delivering resources is necessary for drug delivery. This experiment also compared attractive and repulsive control with the global control used for the other experiments. The experiment challenged players to collect particles using a swarm of 100 robots and return the particles to a home region. The robots encapsulate the particles on contact.

The results were conclusive, as shown in Fig. 8.b: global control was the fastest, followed by repulsive control, with attractive control last. Using ANOVA analysis, we reject the null hypothesis that all controllers are equivalent, with p-value 2.96×10^{-6}.

4 Theory

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:

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

and our state space representation is:

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

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

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

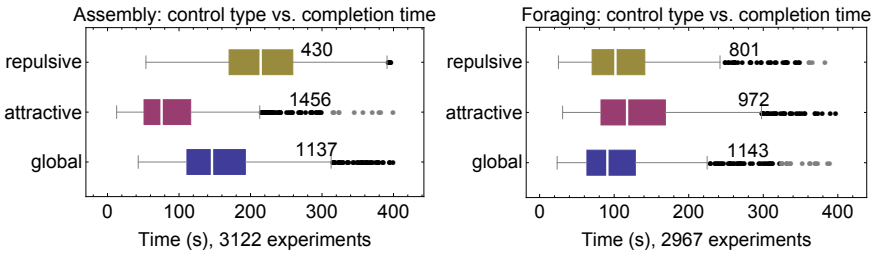


Figure 8: Completion time depends on both the task and the control type. Left: Attractive control resulted in the shortest completion time and repulsive the longest for building a three-block pyramid. Right: in the foraging test, global control resulted in the shortest completion time and attractive the longest.

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

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

4.1 Independent control of many robots is impossible

A single robot is fully controllable. 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. Without loss of generality, assume $m = 1$. Our state-space representation is:

$$\begin{bmatrix} \dot{x}_1 \\ \ddot{x}_1 \\ \vdots \\ \dot{x}_n \\ \ddot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & \dots & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ \dot{x}_1 \\ \vdots \\ x_n \\ \dot{x}_n \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u_x \tag{5}$$

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

$$\mathcal{C}_n = \left[\begin{array}{c|c|c|c|c} 0 & 1 & 0 & 0 & \\ 1 & 0 & 0 & 0 & \\ \vdots & \vdots & \vdots & \vdots & \dots \\ 0 & 1 & 0 & 0 & \\ 1 & 0 & 0 & 0 & \end{array}\right], \quad \text{rank}(\mathcal{C}_n) = 2. \tag{6}$$

4.2 Controlling the mean position of many robots

This means *any* number of robots 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 robots, but have options on which states to control. One option is to control the position and velocity of the

j^{th} robot. To find a potentially more useful option, we create the following reduced order system that represents the average x position and velocity of the n robots:

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

Thus:

$$\begin{bmatrix}\dot{\bar{x}}\\ \ddot{\bar{x}}\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}\bar{x}\\ \dot{\bar{x}}\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix} u_x \tag{7}$$

We again analyze the controllability matrix $\mathcal{C}_\mu$:

$$\mathcal{C}_\mu = \left[\begin{array}{c|c}0 & 1\\ 1 & 0\end{array}\right], \quad \text{rank}(\mathcal{C}_\mu) = 2. \tag{8}$$

Thus the average position and average velocity are controllable.

Due to symmetry of the control input, only the mean position and mean velocity are controllable. However, there are several techniques for breaking symmetry, for example by using obstacles as in Becker et al. (2013), or by allowing independent noise sources as in Becker et al. (2014b).

We control mean position with a PD controller that uses the mean position and mean velocity. $[u_x, u_y]^\top$ is the global force applied to each robot:

$$\begin{aligned} u_x &= K_p(x_{goal} - \bar{x}) + K_d(0 - \dot{\bar{x}})\\ u_y &= K_p(y_{goal} - \bar{y}) + K_d(0 - \dot{\bar{y}}) \end{aligned} \tag{9}$$

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

4.3 Controlling the variance of many robots

The variance, σ_x^2, σ_y^2, of n robots' position is computed as:

$$\begin{aligned} \overline{x}(\mathbf{x}) &= \frac{1}{n}\sum_{i=1}^{n} x_i, \qquad \sigma_x^2(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2,\\ \overline{y}(\mathbf{x}) &= \frac{1}{n}\sum_{i=1}^{n} y_i, \qquad \sigma_y^2(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \overline{y})^2. \end{aligned} \tag{10}$$

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 unmodelled dynamics. These unmodelled dynamics are dominated by Brownian noise, described by Einstein (1956). To model this (1) must be modified as follows:

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) + W\boldsymbol{\varepsilon}(t), \tag{11}$$

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

$$\dot{\sigma}_x^2(\mathbf{x}(t), \mathbf{u}(t) \Leftarrow 0) = W\boldsymbol{\varepsilon}, \quad \sigma_x^2(t) = \sigma_x^2(0) + W\boldsymbol{\varepsilon}t \tag{12}$$

If faster dispersion is needed, the swarm can be pushed through obstacles such as a diffraction grating or Pachinko board as in Becker et al. (2013).

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,

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

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 Sloane (1990) showed that the minimum-variance packing for n circles with radius r is

$$\sigma_{optimal}^2(n, r) \approx \frac{\sqrt{3}}{\pi} nr^2 \approx 0.55nr^2. \tag{14}$$

We will prove the origin is globally asymptotically stabilizable by using a control-Lyapunov function, as in Lyapunov (1992). A suitable Lyapunov function is the squared variance error:

$$\begin{aligned} V(t, \mathbf{x}) &= \frac{1}{2}(\sigma^2(\mathbf{x}) - \sigma_{goal}^2)^2 \\ \dot{V}(t, \mathbf{x}) &= (\sigma^2(\mathbf{x}) - \sigma_{goal}^2)\dot{\sigma}^2(\mathbf{x}) \end{aligned} \tag{15}$$

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

$$u(t) = \begin{cases} \text{move to wall} & \text{if } \sigma^2(\mathbf{x}) > \sigma_{goal}^2 \\ \text{move from wall} & \text{if } \sigma^2(\mathbf{x}) \leq \sigma_{goal}^2. \end{cases} \tag{16}$$

For such a $u(t)$,

$$\dot{\sigma}^2(\mathbf{x}) = \begin{cases} \text{negative} & \text{if } \sigma^2(\mathbf{x}) > \max(\sigma^2_{goal}, \sigma^2_{optimal}(n,r)) \\ W\epsilon & \text{if } \sigma^2(\mathbf{x}) \leq \sigma^2_{goal}, \end{cases} \tag{17}$$

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

A controller to regulate the variance to σ^2_{ref} is:

$$\begin{aligned} u_x &= K_p(x_{goal}(\sigma^2_{ref}) - \bar{x}) - K_d\bar{v}_x + K_i(\sigma^2_{ref} - \sigma^2_x) \\ u_y &= K_p(y_{goal}(\sigma^2_{ref}) - \bar{y}) - K_d\bar{v}_y + K_i(\sigma^2_{ref} - \sigma^2_y). \end{aligned} \tag{18}$$

We call the gain scaling the variance error K_i because the variance, if unregulated, integrates over time. Eq. (18) 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):

$$x_{goal}(\sigma^2_{ref}) = \ell_x/2 = r + \sqrt{3\sigma^2_{ref}} \tag{19}$$

If a wall to the right is closer, the signs of $[K_p, K_i]$ are inverted, and the location x_{goal} is translated.

4.4 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 less than the maximum controlled speed, we can adopt the hybrid, hysteresis-based controller shown in Alg. 1 to regulate the mean and variance. Such a controller normally controls the mean position, but switches to minimizing variance if the variance exceeds some σ^2_{max}. Variance is reduced until less than σ^2_{min}, then control again regulates the mean position. This technique satisfies control objectives that evolve at different rates as in Kloetzer and Belta (2007), and the hysteresis avoids rapid switching between control modes. The process is illustrated in Fig. 9.

A key challenge is to select proper values for σ^2_{min} and σ^2_{max}. The optimal packing variance was given in (14). The random packings generated by pushing our robots into corners are suboptimal, so we choose the conservative values shown in Fig. 9:

$$\begin{aligned} \sigma^2_{min} &= 2.5r + \sigma^2_{optimal}(n,r) \\ \sigma^2_{max} &= 15r + \sigma^2_{optimal}(n,r). \end{aligned} \tag{20}$$

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 $[x_{target}, y_{target}]$.

$flag_x \leftarrow$ **false**, $flag_y \leftarrow$ **false** ▷ initially not in variance control mode
$x_{goal} \leftarrow x_{target}$, $y_{goal} \leftarrow y_{target}$
loop
  if $\sigma_x^2 > \sigma_{max}^2$ **then**
    $x_{goal} \leftarrow x_{min}$
    $flag_x \leftarrow$ **true**
  else if $flag_x$ **and** $\sigma_x^2 < \sigma_{min}^2$
    $x_{goal} \leftarrow x_{target}$
    $flag_x \leftarrow$ **false**
  end if
  if $\sigma_y^2 > \sigma_{max}^2$ **then**
    $y_{goal} \leftarrow y_{min}$
    $flag_y \leftarrow$ **true**
  else if $flag_y$ **and** $\sigma_y^2 < \sigma_{min}^2$
    $y_{goal} \leftarrow y_{target}$
    $flag_y \leftarrow$ **false**
  end if
  Apply (9) to move toward $[x_{goal}, y_{goal}]$
end loop

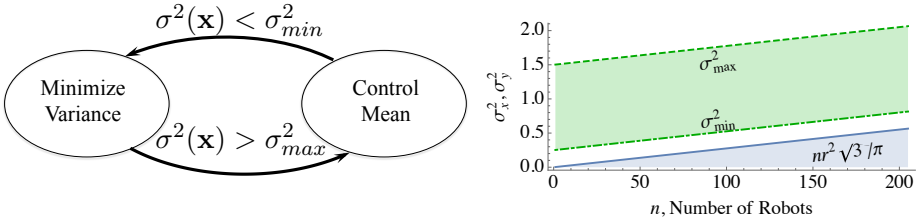


Figure 9: (Left) Two states to control mean and variance of a robot swarm. (Right) Switching conditions for variance control are set as a function of n, and designed to be larger than the optimal packing density. The plot uses robot radius $r = 1/10$.

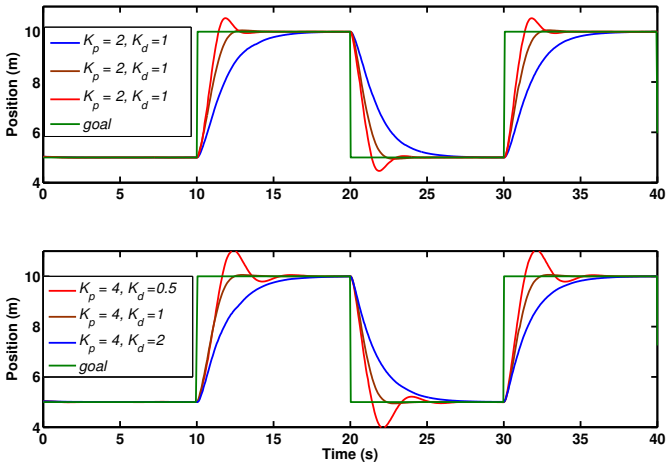


Figure 10: In simulation, tuning proportional (K_p, top) and derivative (K_d, bottom) gain values in (9) improves performance with $n = 100$ robots.

5 Simulation of control laws

Our simulations use a Javascript port of Box2D, a popular 2D physics engine with support for rigid-body dynamics and fixed-time step simulation, presented in Catto (2010). All experiments ran on a Chrome web browser on a 2.6 GHz Macbook. All code is available at Shahrokhi and Becker (2016a).

5.1 Controlling the mean position of many robots

We performed a parameter sweep using the PD controller (9) to identify the best control gains. Representative experiments are shown in Fig. 10. 100 robots were used and the maximum speed was 3 meters per second. As shown in Fig. 10, we can achieve an overshoot of 1% and a rise time of 1.52 s with $K_p = 4$, and $K_d = 1$.

5.2 Controlling the variance of many robots

For variance control we use the control law (18). Results are shown in Fig. 11, with $K_{p,i,d} = [4, 1, 1]$.

5.3 Hybrid control of mean and variance of many robots

Fig. 12 shows a simulation run of the hybrid controller in Alg. 1 with 100 robots in a square workspace containing no internal obstacles.

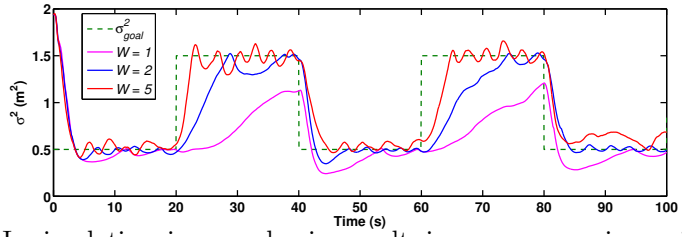


Figure 11: In simulation, increased noise results in more responsive variance control because stronger Brownian noise causes a faster increase of variance.

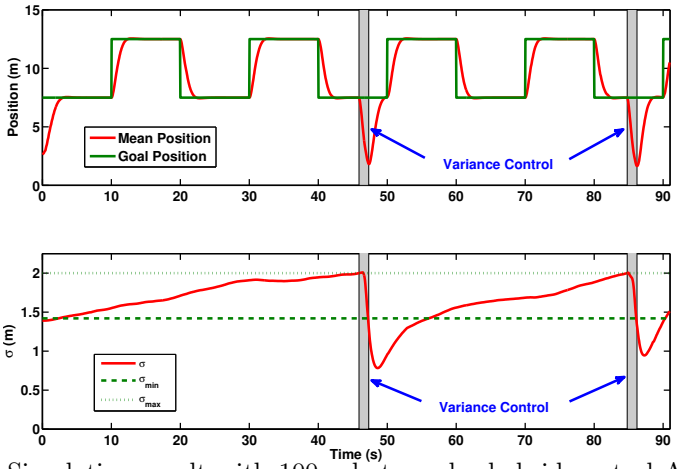


Figure 12: Simulation result with 100 robots under hybrid control Alg. 1, which controls both the mean position (top) and variance (bottom). For ease of analysis, only x position and variance are shown.

6 Object manipulation results

This section analyzes an *object manipulation* task attempted by our hybrid, hysteresis-based controllers, inspired by the analysis of human users in §3.2. The swarm must deliver the object to the goal region. To solve this object manipulation task, we divide the task into two components: designing a policy for the object, and designing a control law for the swarm to push the object according to the policy.

6.1 Learning a policy for the object

To design the policy, we first discretize the environment. In Shahrokhi and Becker (2015), we used breadth-first search (BFS) on this discretized grid to determine $\mathbf{M}_{BFS}$, the shortest distance from any grid cell to the goal, and generated a gradient map $\nabla\mathbf{M}$ shown in Fig. 13. The object's center of mass is at $\mathbf{b}$ and has average radius r_b, so we define the desired direction for the object as $\mathbf{D}(\mathbf{b}) = \nabla\mathbf{M}(\mathbf{b})$. The robots were directed behind the object at $\mathbf{b} - 1.5r_b\mathbf{D}(\mathbf{b})$, then directed to $\mathbf{b} - 0.1r_b\mathbf{D}(\mathbf{b})$ to push the object toward the goal location. This approach using workspace BFS fails to account for the hull of the object and will suggest moves that can cause at least partial collision with the workspace. A configuration-space BFS approach avoids that problem but still fails to model uncertain actuation of the object by the swarm.

To solve both these problems, this paper models object movement as a Markov Decision Process (MDP) with non-deterministic movement. Value iteration, as described in Thrun et al. (2005), is used to learn an *optimal policy*: a function that assigns a control input to every possible state. 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.

A corresponding reward function gives a high reward to the goal state, a large negative reward to states including obstacles, and a small negative reward to all other states. The reward function $r(x, \mathbf{u})$ is defined as

$$r(x,u) = \begin{cases} +100, & \text{if } \mathbf{u} \text{ leads to goal state} \\ -100, & \text{if } \mathbf{u} \text{ leads to a state with an obstacle} \\ -1, & \text{otherwise} \end{cases} \tag{21}$$

$\mathbf{u}$ is the action and x is the current state. Value iteration iteratively learns the value $\hat{V}(x)$ for the object being in each state x. The optimal policy is

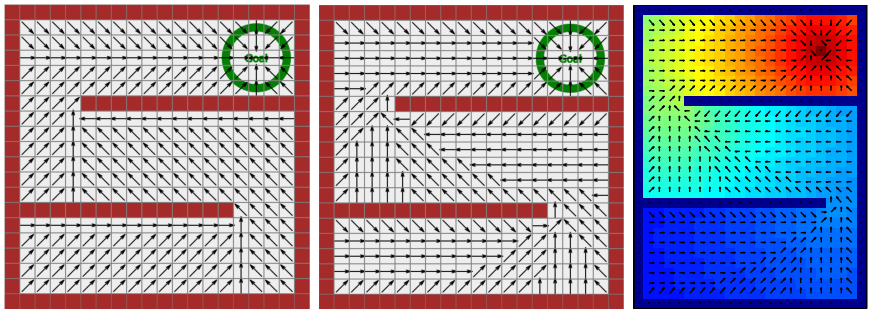


Figure 13: The BFS algorithm finds the shortest path for the moveable object to compute gradient vectors (left). Modeling as an MDP enables encoding penalties for being near obstacles. (Middle) The control policy from value iteration. (Right) The vision algorithm detects obstacles in the hardware setup. This map is used to produce the value function and control policy shown.

the control action that, in probability, results in the largest value:

$$\mathbf{D}(x) = \arg\max_{\mathbf{u}}[r(x,\mathbf{u}) + \sum_{j=1}^{N} \hat{V}(x_j)p(x_j|\mathbf{u},x)]. \tag{22}$$

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

for $i = 1$ to N do

$$\hat{V}(x_j) = \gamma \max_{\mathbf{u}}[r(x_j,\mathbf{u}) + \sum_{i=1}^{N} \hat{V}(x_i)p(x_i|\mathbf{u},x_j)] \tag{23}$$

end

In our experiments $\gamma = 0.97$, and (23) was iterated 200 times. A MATLAB implementation of this algorithm is available at Becker (2015).

$\mathbf{M}_{BFS}$ and the value function are shown in Fig. 13. In 10 simulations with 100 robots, pushing the object to goal using BFS required 183±179 s while Policy Iteration required 90±35 s (mean±std).

6.2 Potential fields for swarm management with a compliant manipulator

Unfortunately, when the swarm is in front of the object, control law (9) pushes the object backwards. To fix this, we implement a potential field approach inspired by Spong and Vidyasagar (2008) that attracts the swarm

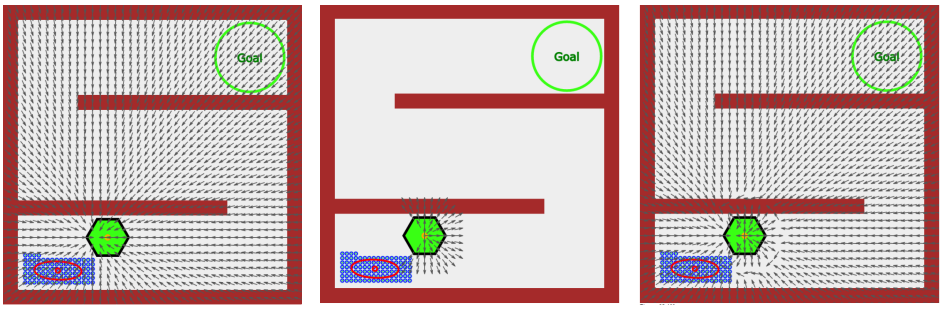


Figure 14: (Left) The attractive field is centered behind the object's COM. (Middle) The repulsive field is centered at the object's COM. (Right) Combining these forces prevents the swarm from pushing the object backwards.

to the intermediate goal, but repulses the swarm from in front of the object. The repulsive potential field is centered at the object's COM and is active for a radius ρ_0, but is implemented only when the swarm mean is within θ of the desired direction of motion $\mathbf{D}(\mathbf{b})$ as shown in Fig. 14.

$$F_{att} = -\zeta \Delta\rho/\rho \tag{24}$$

$$\phi = \cos^{-1}\left(\frac{\mathbf{D}(\mathbf{b}) \cdot ([\bar{x}, \bar{y}] - \mathbf{b})}{\|\mathbf{D}(\mathbf{b})\| \, \|[\bar{x}, \bar{y}] - \mathbf{b}\|}\right) \tag{25}$$

$$F_{rep} = \begin{cases} \eta(1/\rho - 1/\rho_0)\frac{1}{\rho^2}\Delta\rho, & \rho \leq \rho_0 \;\&\; \phi < \theta \\ 0, & \text{otherwise} \end{cases} \tag{26}$$

$$F_{pot} = F_{att} + F_{rep} \tag{27}$$

In simulations, $\theta = \pi/2$, $\eta = 75$, $\zeta = 2$ and $\rho_0 = 3$. Because the kilobot hardware experiments have a slower time constant, they use $\theta = \pi/2$, $\eta = 50$, $\zeta = 1$ and $\rho_0 = 7.5$.

In 10 simulations with 100 robots, pushing the object to goal without a repulsive potential field failed in two of twelve runs. No failures occurred with the repulsive potential field. Of successful trials, completion time without repulsive potential fields required 245±135 s while using repulsive potential fields required 90±35 s (mean±std).

6.3 Outlier rejection

The variance controller in Alg. 1 is a greedy algorithm that is susceptible to outliers. The controller in Shahrokhi and Becker (2016a) failed in 14% trials, often because workspace obstacles made some robots unable to

reach the object. This failure rate increases if object weight increases or ground-robot friction increases. The mean and covariance calculations (10) included all robots in the workspace. Robots that cannot reach the object due to obstacles skew these calculations. The state machine in Fig. 15A 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. 15B shows the regions for the main state. The object is in region 1. An indicator function is applied to (10) so only robots inside region 1 are counted. This filtering increases experimental success because the mean calculation only includes nearby robots 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. 15C. The object is in transfer region 1, so only robots in transfer region 1 are included in the mean and covariance calculations. The robots should push the object to the left. Without filtering using regions, the red circle is the mean and the algorithm would instruct the robots to push the object up. The black circle shows the filtered mean and the algorithm instructs the robots to push the object directly left.

In 10 simulations with 100 robots, completion time without outlier rejection required 245±135 s while using outlier rejection required 90±35 s (mean±std).

6.4 Algorithm for object manipulation

We use the hybrid hysteresis-based controller in Alg. 1 to track the desired position, while maintaining sufficient robot 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 (20). The minimize variance control law (18) 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 object manipulation is listed in Alg. 2.

In rare cases during simulations the swarm may become trapped in a local minimum of (27). If the swarm mean position does not change for five seconds, 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.

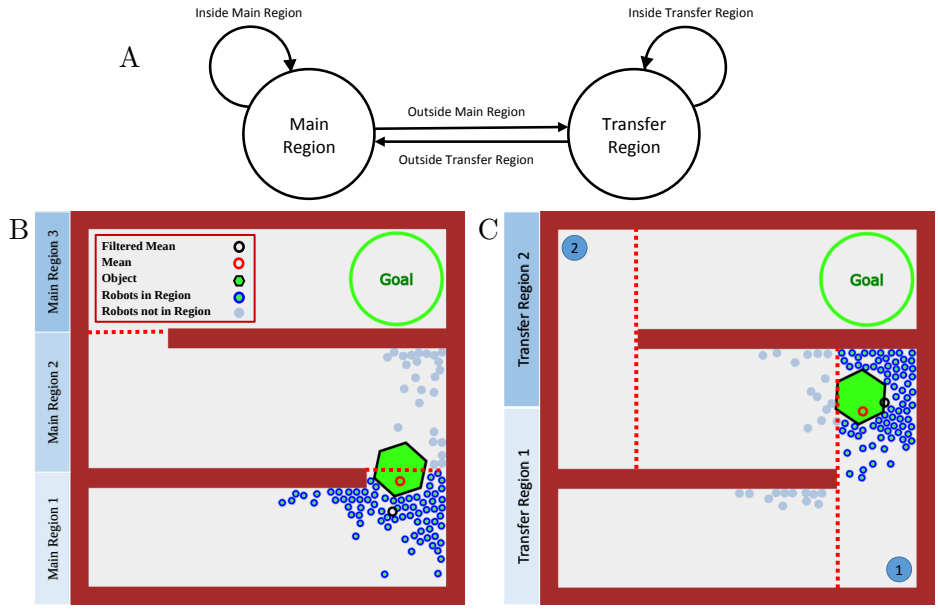
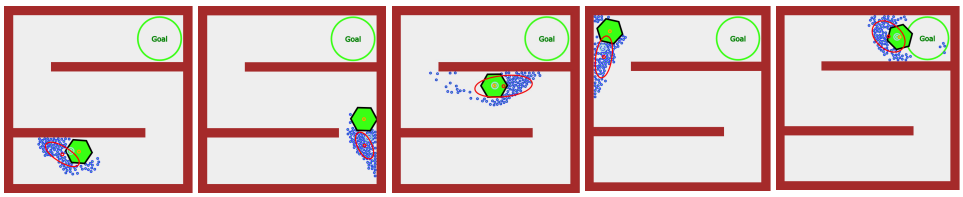


Figure 15: Outlier rejection state machine and regions.

Algorithm 2 Object-manipulation controller for a robotic swarm.

Require: Knowledge of moveable object's center of mass $\mathbf{b}$; swarm mean $[\bar{x}, \bar{y}]$ and variance $[\sigma_x^2, \sigma_y^2]$, each calculated using the regions function from §6.3; map of the environment

Compute optimal policy for object, according to §6.1
while $\mathbf{b}$ is not in goal region **do**
    $\sigma^2 \leftarrow \max(\sigma_x, \sigma_y)$
    if $\sigma^2 > \sigma_{max}^2$ **then**
        while $\sigma^2 > \sigma_{min}^2$ **do**
            $[x_{goal}, y_{goal}] \leftarrow$ the nearest corner in current region
            Apply (9) to move toward $[x_{goal}, y_{goal}]$
        end while
    else
        Calculate $\mathbf{D}(\mathbf{b})$ ▷ direction for object at $\mathbf{b}$
        Apply (27) ▷ potential field to push object in correct direction
    end if
end while



$t = 5$ s $t = 45$ s $t = 60$ s $t = 85$ s $t = 100$ s

Figure 17: Snapshots showing an object manipulation simulation with 100 robots under automatic control. See video in multimedia extension.

6.5 Automated object manipulation (simulation)

Fig. 17 shows snapshots during an execution of this algorithm in simulation. Experimental results of parameters sweeps are summarized in Fig. 18. Each trial measured the time to deliver the object to the goal location. The default parameter settings used 100 robots, a normalized weight of 1, a hexagon shape, and Brownian noise (applied once each simulation step) with $W = 5$.

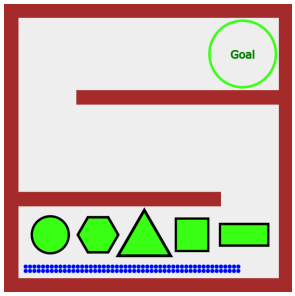


Figure 16: Six equal-area objects were tested.

The interaction between the robots and object is impulsive so, like the study of impulsive pulling in Christensen et al. (2016), adding additional robots decreases completion time, but with diminishing returns. After 75 robots, additional robots no longer can interact with the object and do not contribute to the task success. 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 robots 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.

7 Object manipulation with hardware robots

Our experiments use centimeter-scale hardware systems called *kilobots*. While those are far larger than the micro scale 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, reported in Rubenstein et al. (2012, 2014), is a low-cost robot de-

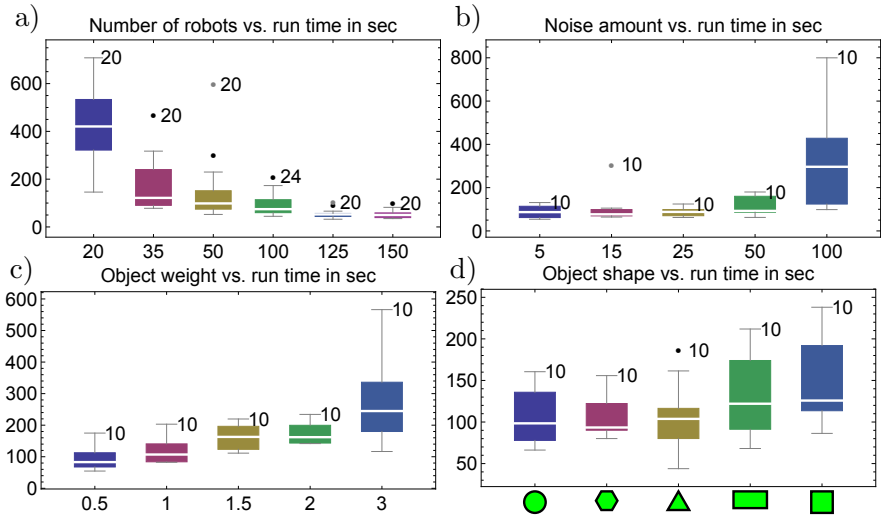


Figure 18: Parameter sweep for a) number of robots, b) different noise values, c) object weight, and d) object shape. Each bar is labelled with the number of trials. Completion time is in seconds.

signed for testing collective algorithms with large numbers of robots. It is available as an open-source platform or commercially from K-Team (2015). 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.

7.1 Environmental setup

In these experiments as shown in Fig. 19, we used n=100 kilobots, a 1.5 m×1.2 m whiteboard as the workspace, and lights: four 50W LED floodlights at the corners and four 30W LED floodlights on the sides of a 6 m square centered on the workspace and 1.5 m above the table. An Arduino Uno connected to an 8-relay shield controlled the lights.

Above the table, an overhead machine vision system tracks the swarm. The vision system identifies obstacles by color segmentation, determines the corners (used to decrease variance), the object by color segmentation, and identifies robots using color segmentation and circle detection with a circular Hough transform.

The objects were 3D printed from 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 (18 cm×27 cm) rectangle, all

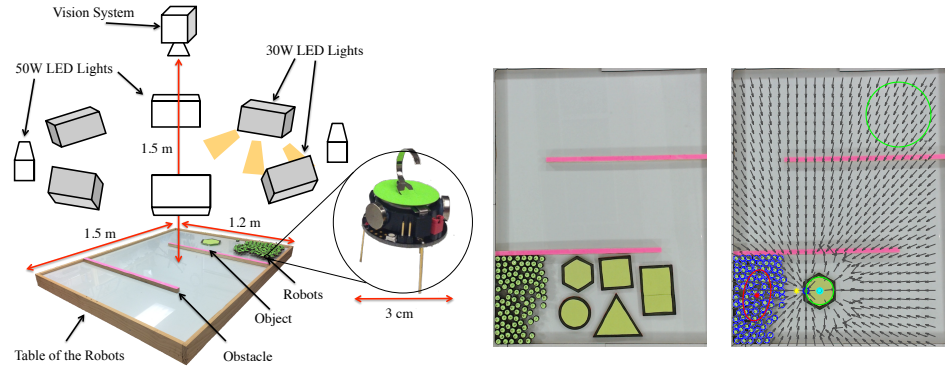


Figure 19: Hardware platform. At right are the shapes used for hardware experiments and a visualization of the potential field.

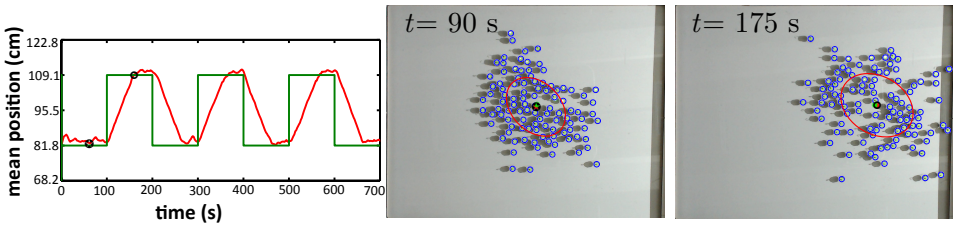


Figure 20: Regulating average x position of 100 kilobots using control law (9).

shown in Fig. 19.The laser-cut patterns for the neon green fiducial markers on the robots and 3D files for objects are available at our github repository, Shahrokhi and Becker (2016b).

Swarm mean control (hardware experiment) Unlike the PD controller (9), 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 constant velocity toward this light. The kilobots oscillate along this orientation because they only have one light detector.

We use the sign of (9), and choose the closest orientation to $\mathbf{D}(\mathbf{b})$ among the eight light sources. Fig. 20 shows that this limited, discretized control still enables regulating the mean position of a swarm of 100 robots.

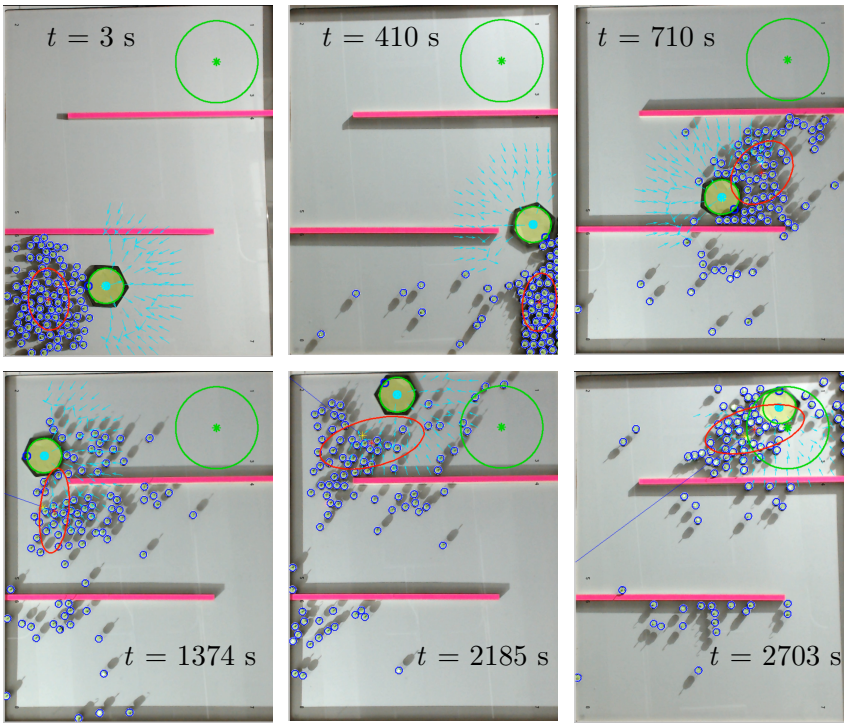


Figure 21: Snapshots showing the object manipulation experiment with 100 kilobots under automatic control. The automatic controller classifies pink objects as obstacles and generates a policy to the goal. See video in multimedia extension.

7.2 Automated object manipulation (hardware experiment)

The kilobots performed five successful runs of object manipulation through the obstacle maze. Each trial used 100 kilobots. Videos of each are in the multimedia extension. Trials two-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. 21. The moveable object was placed in the lower center of the workspace. MATLAB code for vision processing, the policy iteration of §6.1 and the algorithm of §6.4 is available on MATLAB Central by Shahrokhi et al. (2016). 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±std). A circular object completed in 3155 s. A square object completed in 6871 s. A triangular object completed in ??:?? minutes.

8 Conclusion

Micro- and nanorobotics have the potential to revolutionize many applications including targeted material delivery, assembly, and surgery. Their small size makes individual control and autonomy challenging, so currently these robots 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 game play results revealed benefits of measuring and controlling statistics of the swarm rather than full state feedback.

Inspired by human operators, this paper designed controllers and controllability results using only the mean and variance of a robot swarm. These controllers were used as primitives for an object manipulation task. 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.

Manipulation by large populations of robots has many open questions. We invite other collaborators to submit their own experiments for large-scale trials to SwarmControl.net. Topics of interest include control with nonuniform flow such as fluid in an artery, gradient control fields like that of an MRI, competitive playing, multi-modal control, optimal-control, and targeted drug delivery in a vascular network.

Funding

This work was supported by the National Science Foundation under Grant No. [IIS-1553063].

References

Michael D. Armani, Satej V. Chaudhary, Roland Probst, and Benjamin Shapiro. Using feedback control of microflows to independently steer multiple particles. *Journal of Microelectromechanical systems*, 15(4):945–956, August 2006.

Phaedon Avouris. Manipulation of matter at the atomic and molecular levels. *Accounts of chemical research*, 28(3):95–102, 1995.

Shishir Bashyal and Ganesh Kumar Venayagamoorthy. Human swarm interaction for radiation source search and localization. In *Swarm Intelligence Symposium (SIS)*, pages 1–8. IEEE, 2008.

Aaron T. Becker. "MDP robot grid-world example", https://www.mathworks.com/matlabcentral/fileexchange/49992-mdp-robot-grid-world-example, March 2015.

Aaron T. Becker and Timothy Bretl. Approximate steering of a unicycle under bounded model perturbation using ensemble control. *IEEE Trans. Robot.*, 28(3):580–591, June 2012.

Aaron T. Becker, Robert Sandheinrich, and Timothy Bretl. Automated manipulation of spherical objects in three dimensions using a gimbaled air jet. In *IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS)*, pages 781–786, Oct. 2009.

Aaron T. Becker, Cem Onyuksel, and Timothy Bretl. Feedback control of many differential-drive robots with uniform control inputs. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2256–2262, October 2012.

Aaron T. Becker, Golnaz Habibi, Justin Werfel, Michael Rubenstein, and James 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)*, pages 520–527, November 2013.

Aaron T. Becker, Chris Ertel, and James McLurkin. Crowdsourcing swarm manipulation experiments: A massive online user study with large swarms of simple robots. In *IEEE International Conference on Robotics and Automation*, pages 2825–2830, 2014a.

Aaron T. Becker, Cem Onyuksel, Timothy Bretl, and James McLurkin. Controlling many differential-drive robots with uniform control inputs. *The International Journal of Robotics Research*, 33(13):1626–1644, 2014b.

Erin Catto. User manual, Box2D: A 2D physics engine for games, http://www.box2d.org, 2010. URL `http://www.box2d.org`.

Jack Cazes. *Encyclopedia of Chromatography*, volume 2. Taylor & Francis Group, New York, second edition, 2005. ISBN 0824727878.

Pinn-Tsong Chiang, Johannes Mielke, Jazmin Godoy, Jason M. Guerrero, Lawrence B. Alemany, Carlos J. Villagómez, Alex Saywell, Leonhard Grill, and James M. Tour. Toward a light-driven motorized nanocar: Synthesis and initial imaging of single molecules. *ACS Nano*, 6(1):592–597, February 2011. doi: 10.1021/nn203969b.

David L Christensen, Srinivasan A Suresh, Katie Hahm, and Mark R Cutkosky. Lets all pull together: Principles for sharing large loads in microrobot teams. *IEEE Robotics and Automation Letters*, 1(2):1089–1096, 2016.

Jean-Pierre de la Croix and Magnus Egerstedt. Controllability characterizations of leader-based swarm interactions. In *AAAI Fall Symposium Series*, 2012.

Raphael Deimel and Oliver Brock. A novel type of compliant, underactuated robotic hand for dexterous grasping. *Robotics: Science and Systems, Berkeley, CA*, pages 1687–1692, 2014.

Eric Diller, Joshua Giltinan, and Metin Sitti. Independent control of multiple magnetic microrobots in three dimensions. *The International Journal of Robotics Research*, 32(5):614–631, 2013. URL `http://ijr.sagepub.com/content/32/5/614.abstract`.

Bruce R. Donald, Christopher G. Levey, Craig D. McGray, Igor Paprotny, and Daniela Rus. An untethered, electrostatic, globally controllable MEMS micro-robot. *J. of MEMS*, 15(1):1–15, February 2006. ISSN 1057-7157.

Bruce R. Donald, Christopher G. Levey, and Igor Paprotny. Planar microassembly by parallel actuation of MEMS microrobots. *J. of MEMS*, 17(4):789–808, August 2008.

Albert Einstein. *Investigations on the Theory of the Brownian Movement*. Courier Corporation, 1956.

Chris Ertel, Shiva Shahrokhi, and Aaron T. Becker. SwarmControl.net git repository, August 2016. URL `https://github.com/RoboticSwarmControl/SwarmControlRedux.git`.

J. Fink, M.A. Hsieh, and V. Kumar. Multi-robot manipulation via caging in environments with obstacles. In *Robotics and Automation, ICRA IEEE International Conference on*, pages 1471–1476, May 2008.

Steven Floyd, Eric Diller, Chytra Pawashe, and Metin Sitti. Control methodologies for a heterogeneous group of untethered magnetic micro-robots. *Int. J. Robot. Res.*, 30(13):1553–1565, November 2011.

Dominic Frutiger, Bradley Kratochvil, Karl Vollmers, and Bradley J. Nelson. Magmites - wireless resonant magnetic microrobots. In *IEEE Int. Conf. Rob. Aut.*, pages 1770–1771, Pasadena, CA, May 2008.

Michael A Goodrich, Sean Kerman, Brian Pendleton, and PB Sujit. What types of interactions do bio-inspired robot swarms and flocks afford a human? In *Robotics: Science and Systems*, 2012.

Ronald L. Graham and Neil JA Sloane. Penny-packing and two-dimensional codes. *Discrete & Computational Geometry*, 5(1):1–11, 1990.

Sariel Har-Peled. On the expected complexity of random convex hulls. *arXiv preprint arXiv:1111.5340*, 2011.

K-Team. Kilobot, www.k-team.com/mobile-robotics-products/kilobot, 2015.

Islam S. M. Khalil, Frank van den Brink, Ozlem Sardan Sukas, and Sarthak Misra. Microassembly using a cluster of paramagnetic microparticles. In *IEEE International Conference on Robotics and Automation*, pages 5507–5512, Karlsruhe, Germany, May 2013.

Marius Kloetzer and Calin Belta. Temporal logic planning and control of robotic swarms by hierarchical abstractions. *Robotics, IEEE Transactions on*, 23(2):320–330, 2007.

Andreas Kolling, Steven Nunnally, and Michael Lewis. Towards human control of robot swarms. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pages 89–96. ACM, 2012.

Aleksandr Mikhailovich Lyapunov. *The General Problem of the Stability of Motion, translated and edited by A.T. Fuller.* Tayor & Francis, London, 1992.

Kevin M. Lynch. Locally controllable manipulation by stable pushing. *IEEE Trans. Robot. Autom.*, 15(2):318–327, April 1999.

Sylvain Martel, Samira Taherkhani, Maryam Tabrizian, Mahmood Mohammadi, Dominic de Lanauze, and Ouajdi Felfoul. Computer 3d controlled bacterial transports and aggregations of microbial adhered nanocomponents. *Journal of Micro-Bio Robotics*, 9(1-2):23–28, 2014.

Lael U. Odhner, Leif P. Jentoft, Mark R. Claffee, Nicholas Corson, Yaroslav Tenzer, Raymond R. Ma, Martin Buehler, Robert Kohout, Robert D. Howe, and Aaron M. Dollar. A compliant, underactuated hand for robust manipulation. *The International Journal of Robotics Research*, 33(5):736–752, 2014.

Dan R. Olsen Jr. and Stephen Bart Wood. Fan-out: Measuring human control of multiple robots. In *SIGCHI Conference on Human Factors in Computing Systems*, pages 231–238, Vienna, Austria, April 2004.

Kathrin E. Peyer, Li Zhang, and Bradley J. Nelson. Bio-inspired magnetic swimming microrobots for biomedical applications. *Nanoscale*, 2013.

Michael Rubenstein, Christian Ahler, and Radhika Nagpal. Kilobot: A low cost scalable robot system for collective behaviors. In *IEEE Int. Conf. Rob. Aut.*, pages 3293–3298, May 2012.

Michael Rubenstein, Alejandro Cornejo, and Radhika Nagpal. Programmable self-assembly in a thousand-robot swarm. *Science*, 345(6198): 795–799, 2014.

Shiva Shahrokhi and Aaron T. Becker. Stochastic swarm control with global inputs. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 421–427, September 2015.

Shiva Shahrokhi and Aaron T. Becker. BlockPushingIROS2015, https://github.com/aabecker/swarmcontrolsandbox/blob/master/ examplecontrollers/blockpushingiros2015.html, July 2016a.

Shiva Shahrokhi and Aaron T. Becker. "Wall-Friction Swarm Shape Control", https://github.com/aabecker/swarmcontrolsandbox/tree/master/papers/icra2016, August 2016b.

Shiva Shahrokhi, Lillian Lin, Mable Wan, and Aaron T. Becker. "Kilobot Swarm Control using Matlab + Arduino", https://www.mathworks.com/matlabcentral/fileexchange/58690, August 2016.

Yasuhiro Shirai, Andrew J. Osgood, Yuming Zhao, Kevin F. Kelly, and James M. Tour. Directional control in thermally driven single-molecule nanocars. *Nano Letters*, 5(11):2330–2334, February 2005. doi: 10.1021/nl051915k.

Mark W. Spong and Mathukumalli Vidyasagar. *Robot dynamics and control*. John Wiley & Sons, 2008.

Peter Squire, Greg Trafton, and Raja Parasuraman. Human control of multiple unmanned vehicles: effects of interface type on execution and task switching times. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, HRI '06, pages 26–32, New York, NY, USA, 2006. ACM. ISBN 1-59593-294-1. doi: 10.1145/1121241.1121248.

Attawith Sudsang, Fred Rothganger, and Jean Ponce. Motion planning for disc-shaped robots pushing a polygonal object in the plane. *IEEE Trans. Robot. Autom.*, 18(4):550–562, August 2002.

Kiyonori Takahashi, Naoko Ogawa, Hiromasa Oku, and Koichi Hashimoto. Organized motion control of a lot of microorganisms using visual feedback. In *IEEE Int. Conf. Rob. Aut.*, pages 1408–1413, May 2006. doi: 10.1109/ROBOT.2006.1641906.

Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, September 2005.

Soichiro Tottori, Li Zhang, Famin Qiu, Krzysztof K Krawczyk, Alfredo Franco-Obregón, and Bradley J Nelson. Magnetic helical micromachines: Fabrication, controlled swimming, and cargo transport. *Advanced Materials*, 24(811):811–816, 2012.